HARVARD UNIVERSITY

DEPARTMENT OF STATISTICS

(4)

AD-A220 416

The Trickett-Welch "Solution" to the Behrens-Fisher Problem Applied to One-sided Tolerance Limits for Random Effects Models

Mark G. Vangel

U.S. Army Materials Technology Laboratory
Watertown, MA 02171-0001

and

Department of Statistics
Harvard University
Cambridge, MA 02138

Technical Report No. ONR-C-5
January 1990

90 04 11 110

Abstract

Let X be a normally distributed random variable with mean μ and variance $\sigma^2 = \sigma_b^2 + \sigma_e^2$. A lower confidence limit for a quantile of this population (i.e., a tolerance limit) is to be determined using data from a one-way balanced random effects ANOVA sample with between-group and within-group variances σ_b^2 and σ_e^2 respectively.

For example, let X represent the strength of a randomly selected specimen of a material manufactured in a batch which can be considered to be randomly selected from a population of batches. A quantity of interest to aircraft designers is the 'B-basis value', which is a 95 percent lower confidence limit on the tenth percentile of the distribution of X. For this situation, it is important that nearly the nominal coverage probability be attained whatever the unknown population variance ratio. It is also very desirable that the calculated limit be as large as possible, since unnecessarily low values cause undue conservatism in design.

This problem is closely related to the Behrens-Fisher problem. We have in the one-way ANOVA two independent mean squares with expected values equal to linear combinations of the variance components, while for the two sample problem the expected values are the sample variances. The Welch series approach (Welch, 1947) can be applied here to produce an approximation which is adequate for many batches. A little known paper by Trickett and Welch (1954) describes an equivalent integral equation which is applied to the tolerance limit problem with dramatic results. Unlike the Welch series calculations, which are as tedious to do today as they were forty years ago, the Trickett-Welch approach is numerical, and one can calculate the successive approximations to orders inconceivable in 1954. In fact, though there is strictly speaking no 'well behaved' exact solution to this problem, one can get amazingly close to the nominal coverage probability for any value of the nuisance parameter by beginning with the first order Welch approximation and iterating an improvement of the Trickett-Welch procedure numerically.

A solution due to Mee and Owen (1983) based on Satterthwaite's (1946) approximation for the distribution of a linear combination of χ^2 random variables is compared with the above approach and with the solution for known variance ratio. The comparison is made both in terms of the coverage probability and by calculation of the probability distributions of the tolerance limits.

1 Introduction

If a material is manufactured in many large batches and the population of interest consists of all batches, a random effects model may be an appropriate model for measurements made on characteristics of the material.

Let X_{ij} denote the jth of J observations from the ith of I batches. If X_{ij} follows a one-way balanced random-effects model, then

$$X_{ij} = \mu + b_i + e_{ij}, \tag{1}$$

where μ denotes the population mean, $\mu + b_i$ denotes the mean of the ith batch, and e_{ij} is the error term. The b_i's and the e_{ij}'s are assumed to be independently distributed normal with mean zero and variance σ_b^2 and σ_e^2 respectively. An observation X from this population is thus normally distributed with mean μ and variance

$$\sigma_X^2 = \sigma_b^2 + \sigma_e^2. \tag{2}$$

This paper presents techniques for determining one-sided tolerance limits for X based on a random sample of J items from each of I batches. A (β, γ) lower tolerance limit is a random variable T such that at least a proportion β of the population is covered by the interval (T, ∞) with probability at least γ. The methods developed here for lower tolerance limits can be adapted in an obvious way to upper limits. We will refer to β as the *coverage* and γ as the *coverage probability.*

An important industrial application of tolerance limits is to the characterization and certification of structural materials for aircraft. In order to determine the acceptability of material for aircraft applications, designers use 'material basis properties' which are tolerance limits on the strength of a material as determined from experimental failure data. A (.90, .95) lower tolerance limit is called a 'B-basis' value or 'B-allowable'. The more stringent (.99, .95) limit is referred to as an 'A-basis' value or 'A-allowable'.

There is increasing interest in the use of composite materials as lightweight alternatives to metals for aircraft applications. Composite material properties typically exhibit far more batch-to-batch variability than do metals; consequently there is a growing need for methods to determine one sided tolerance limits in the presence of batch-to-batch variation.

Various approaches to this tolerance limit problem will be discussed below. The ratio of population variance components is a nuisance parameter for this problem. How one chooses to address this complication is a distinguishing feature of the alternative methods.

Section 2 presents a method due to Mee and Owen (1983) based on Satterthwaite's (1946) approximate distribution for a linear combination of mean squares. A modification is suggested which slightly improves on the Mee-Owen result.

In Section 3 the exact solution is derived for known variance ratio in terms of a generalization of the noncentral t-distribution. This distribution is used in Section 4 to examine the effect on the coverage probability γ of pooling and using a simple random sample procedure when the variance ratio is not zero. Section 5 consists of an asymptotic series solution (following Welch (1947)) for the tolerance limit limit factor to terms of $O(1/n)$. In Section 6 this problem is formulated as an integral equation and a method due to Trickett and Welch (1954) is applied. Following the improvements of Section 7, this approach is shown to provide virtually exact tolerance limits for the case of an unknown nuisance parameter.

The Trickett-Welch approach has received little attention in the statistics literature. The dramatic success of this numerical method for the problem considered in this paper suggests that one might profitably apply this technique to a large class of inference problems. Two such potential applications are outlined in Section 8.

The cumulative distribution function of the tolerance limits are examined in Section 9. These cdfs may be easily calculated in terms of the generalized noncentral t-distribution of Section 3.

The discussion of the various tolerance limit procedures in Section 10 makes use of both the calculated coverage probability as a function of the nuisance parameter and the cdfs of the tolerance limits in making comparisons.

Finally, a simulated data example is considered in Section 11.

2 The Mee-Owen Procedure

Let $n = IJ$ denote the sample size. The parameters μ, σ_e^2 and σ_b^2 of the random effects model may be estimated by the pooled mean $\hat{\mu}$, the within batch mean square MS_e, and a linear combination of MS_e with the between batch mean square MS_b where:

$$\hat{\mu} = \sum_{i=1}^{I} \sum_{j=1}^{J} \frac{X_{ij}}{IJ}, \tag{3}$$

$$MS_b = J\sum_{i=1}^{I}\frac{(\hat{\mu}-\bar{X}_i)^2}{I-1}, \tag{4}$$

and

$$MS_e = \sum_{i=1}^{I}\sum_{j=1}^{J}\frac{(X_{ij}-\bar{X}_i)^2}{I(J-1)}. \tag{5}$$

An unbiased estimator of the population variance σ_X^2 is

$$\hat{\sigma}_X^2 = MS_b/J + (1-1/J)MS_e. \tag{6}$$

For $0 < \beta < 1$, let K_β be the β quantile of the standard normal distribution, i.e

$$\beta = \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{K_\beta} e^{-t^2/2}dt. \tag{7}$$

A (β,γ) lower tolerance limit is a 100γ percent lower confidence bound for

$$\mu - K_\beta\sigma_X. \tag{8}$$

By analogy with the single sample case (see, for example, Owen (1968)), one seeks an estimator of the form

$$\hat{\mu} - k\hat{\sigma}_X, \tag{9}$$

where k is chosen to satisfy

$$P(\hat{\mu} - k\hat{\sigma}_X \leq \mu - K_\beta\sigma_X) = \gamma. \tag{10}$$

Since $\hat{\mu}$ is distributed normal with mean μ and variance

$$\sigma_{\hat{\mu}}{}^2 = (J\sigma_b^2 + \sigma_e^2)/n, \tag{11}$$

one may rewrite (10) as

$$P\left(\frac{Z+\sqrt{n}K_\beta B}{\hat{\sigma}_X/\sigma_X} \leq \sqrt{n}kB\right) = \gamma, \tag{12}$$

where

$$Z \equiv \frac{\hat{\mu}-\mu}{\sigma_{\hat{\mu}}}, \tag{13}$$

$$B \equiv \sqrt{\frac{R+1}{JR+1}}, \tag{14}$$

and

$$R \equiv \sigma_b^2/\sigma_e^2. \tag{15}$$

The random variable $(\hat{\sigma}_X^2/\sigma_X^2)$ is approximately distributed as the ratio of a χ^2 to its degrees of freedom, where the degrees of freedom are given by Satterthwaite's (1946) approximation:

$$f = \frac{(R+1)^2}{\frac{(R+1/J)^2}{I-1} + \frac{1-1/J}{n}}. \tag{16}$$

If $T_f^{-1}(\gamma, \delta)$ denotes the inverse of the noncentral t-distribution with f degrees of freedom and noncentrality parameter δ then one may make the following approximation:

$$k \approx \frac{T_f^{-1}(\gamma, \sqrt{n}K_\beta B)}{\sqrt{n}B}. \tag{17}$$

Unfortunately, this tolerance limit factor k depends on the nuisance parameter R. Mee and Owen (1983) suggest replacing R with

$$\hat{R}_\eta \equiv \frac{F_\eta \frac{MS_b}{MS_e} - 1}{J}, \tag{18}$$

where F_η is the 100η percentile of an F random variable with numerator and denominator degrees of freedom $I(J-1)$ and $I-1$, respectively. $\hat{R}_\eta$ is a 100η percent upper confidence bound estimate for R (Searle, 1971, p.414).

Having made the approximation (17), we may determine the coverage probability

$$P(\hat{\mu} - k(\hat{R}_\eta)\hat{\sigma}_X \leq \mu - K_\beta \sigma_X) = \gamma^*(\eta, I, J, R). \tag{19}$$

As R tends to infinity, (17) becomes

$$k = T_{I-1}^{-1}(\gamma, \sqrt{I}K_\beta)/\sqrt{I} \tag{20}$$

for all η and J. The case of infinite R corresponds to $\sigma_e^2 = 0$; the model (1) reduces to a simple random sample of size I, and (20) provides an exact solution. Hence for all η, I, and J

$$\lim_{R \to \infty} \gamma^*(\eta, I, J, R) = \gamma. \tag{21}$$

For R sufficiently small, it can be shown that γ^* exceeds the nominal level γ. However, in general γ^* will be less than γ for an interval of intermediate

R values. It turns out that one can determine $\eta = \eta(\beta, \gamma)$ numerically so that $\gamma^* \geq \gamma$ for all I, J, and R. These η values are reproduced from Mee and Owen (1983) for various combinations of β and γ in Table 1.

The first improvement to this tolerance limit procedure that we will consider is to allow η to vary with I and J. This gives a modified Mee-Owen procedure with coverage probability closer to the nominal value. The result of this numerical work for the case of $\beta = .90$ and $\gamma = .95$ is presented in Table 2.

3 An Exact Solution for Known R

For a simple random sample, a solution to the one sided tolerance limit problem is readily obtained in terms of the noncentral t-distribution. If one assumes that the variance ratio, R, is known, then the corresponding problem for a sample from a balanced random effects model can be solved almost as easily. What is required is the distribution of a 'generalized noncentral t' random variable, a generalization of the noncentral t to a random variable with the square root of a linear combination of two χ^2's in the denominator.

Let $\delta = \sqrt{n} K_\beta B$, $n_1 = I - 1$, and $n_2 = I(J-1)$ where B is defined in (14). If R is known, the tolerance limit factor k is the appropriate quantile of the distribution of

$$A = (n_1 + n_2)^{1/2} \frac{Z + \delta}{\sqrt{d_1 Y_1 + d_2 Y_2}}, \tag{22}$$

where Z has a standard normal distribution; Y_i is distributed as a χ^2 with n_i degrees of freedom for $i = 1, 2$; d_1, d_2, and δ are constants with d_1 and d_2 positive; and Z, Y_1, and Y_2 are mutually independent. Once this distribution has been determined the tolerance limit may be obtained exactly. The cdf of the linear combination $Y \equiv d_1 Y_1 + d_2 Y_2$ is show in Fleiss (1971) to be

$$F_Y(y) = E_\nu \left[\chi^2_{n_1+n_2} \left(y / \left(d_1 X + d_2 (1 - X) \right) \right) \right], \tag{23}$$

where $\chi^2_{n_1+n_2}$ is the chi-square cumulative distribution with f degrees of freedom and the expectation is with respect to a beta random variable, X, with parameter $\nu = (n_1/2, n_2/2)$.

By conditioning on the denominator of (22) one sees that

$$F_A(k) \equiv P(A \leq k) = E_\nu \left[T_{n_1+n_2} \left(k \sqrt{d_1 X + d_2 (1 - X)}, \delta \right) \right], \tag{24}$$

where $T_f(t,\delta)$ denotes the noncentral t cumulative distribution with f degrees of freedom and noncentrality parameter δ, i.e.

$$T_f(t,\delta) = \int_0^\infty \Phi\left(t\sqrt{\frac{w}{f}} - \delta\right) C_f(w)\, dw, \tag{25}$$

where C_f denotes the χ^2 density with f degrees of freedom and $\Phi(\cdot)$ is the standard normal distribution.

For the tolerance limit problem, let

$$d_1 = \frac{(n_1 + n_2)I}{I - 1}, \tag{26}$$

and

$$d_2 = \frac{n_1 + n_2}{JR + 1}. \tag{27}$$

where I, J, K_β, and R are as in Sections 1 and 2 and n_1, n_2, and δ are given above. The value $k(R)$ such that $F_A(k) = \gamma$ thus provides an exact solution to the problem of known R.

Although the above derivation is simple, it is apparently not well known. A much more complicated representation of the distribution of the random variable (22) is developed in Ray and Pitman (1961).

4 The Effect of Pooling on the Coverage Probability

The tolerance limit procedure discussed in Section 2 is conservative (i.e. provides a coverage probability greater than the nominal value) when the population variance ratio, R, is small. Mee and Owen (1983) therefore suggest that data be pooled and that a single sample method be applied when the mean square ratio is less than one. They then proceed to investigate the conditional behavior of their proposed estimator.

Using the distribution developed in Section 3, we shall determine the coverage probability for a single sample procedure applied to pooled data as a function of the variance ratio. This result will be used to determine the unconditional coverage probability of the Mee-Owen method in Section 10.

Let Y_1 and Y_2 be as in (22) and let n_1 and n_2 denote the between and within batch degrees of freedom respectively. The pooled variance estimate is

$$S_p^2 = \frac{\sum_{i=1}^{I} \sum_{j=1}^{J} (X_{ij} - \hat{\mu})^2}{n - 1}, \tag{28}$$

where n denotes the pooled sample size, I the number of batches, J the batch size, and $\hat{\mu}$ the grand mean. Partitioning the total mean square and substituting (11) for the variance of $\hat{\mu}$, one obtains

$$\frac{S_p^2}{\sigma_{\hat{\mu}}{}^2} = \frac{n}{n-1} \frac{\sigma_e^2 Y_2 + (J\sigma_b^2 + \sigma_e^2)Y_1}{J\sigma_b^2 + \sigma_e^2}. \tag{29}$$

If k_0 denotes the single sample tolerance limit factor (e.g. Owen, 1968, pp. 446-448), then the coverage probability as a function of R is

$$\begin{aligned} \hat{\gamma}(R) &= P(\hat{\mu} - k_0 S_p \leq \mu - K_\beta \sigma_X) \\ &= P\left(\frac{Z + \sqrt{n} K_\beta B}{S_p/\sigma_{\hat{\mu}}} \leq k_0\right), \end{aligned} \tag{30}$$

with notation as in Section 2. Substituting (29) into (30) and employing the distribution (24), one may readily examine $\hat{\gamma}(R)$ numerically. For the present application of (24) the constants d_1 and d_2 can easily be determined from inspection of (30); they are different than the values assigned in (26) and (27).

From the typical plot in Figure 1 it is apparent that the coverage probability obtained will be substantially less than the nominal value even for fairly small values of R. Clearly, criteria which result in the decision to pool must be considered carefully if one is to minimize the risk of very anticonservative tolerance limits in the presence of batch-to-batch variation. Alternatively, one might seek an estimator which performs well for all R, eliminating the consideration of pooling altogether. This approach will be taken in Section 6.

5 The Solution for Unknown R: Welch-Aspin Series

For unknown variance ratio, the tolerance limit problem is closely related to the Behrens-Fisher problem. Since it has been shown by Linnik (1968, Ch. 9) that there is no 'well behaved' solution to the Behrens-Fisher problem, it follows that we also are faced with a problem without an exact solution. However, one can proceed as if a solution does exist and attempt to approximate it. Following the work of Welch (1947) and Trickett and Welch (1954), two forms for such an approximate solution are obtained.

A series solution is developed first. While computationally simple, the first order approximation presented here is anticonservative and may only be suitable for many batches.

One could improve this procedure by taking higher order approximations. However, this becomes very tedious to carry out. Alternatively, the tolerance limit factor as a function of the mean square ratio may be obtained approximately as the solution of an integral equation. Although this requires the use of a computer, the method which results appears to give very nearly the nominal coverage probability - even for small sample sizes.

To simplify the notation in what follows, let S_i^2 be the mean squares, σ_i^2 their expected values, and n_i the asociated degrees of freedom for $i = 1, 2$, i.e. :

$$\begin{array}{lll} S_1^2 = MS_b, & \sigma_1^2 = J\sigma_b^2 + \sigma_e^2, & n_1 = I - 1, \\ S_2^2 = MS_e, & \sigma_2^2 = \sigma_e^2, & n_2 = I(J-1). \end{array}$$

The pooled sample size is $n = IJ$ and the population variance is denoted by

$$\sigma^2 = \sigma_X^2 = \sigma_b^2 + \sigma_e^2 = \sigma_1^2/J + \sigma_2^2(1 - 1/J), \tag{31}$$

and estimated by

$$S^2 = \hat{\sigma}_X^2 = S_1^2/J + S_2^2(1 - 1/J). \tag{32}$$

The subscript X for the population variance and estimates of this variance will be omitted for the remainder of this section.

The tolerance limit factor will be denoted k as in (10) , and we define $h(S_1^2, S_2^2)$ to be $k\hat{\sigma}$. The tolerance limit may be expressed as an expectation with respect to the distributions of the mean squares in terms of the standard normal distribution:

$$\begin{aligned} \gamma &= P(\hat{\mu} - k\hat{\sigma} \leq \mu - K_\beta\sigma) \\ &= E\left[\Phi\left(\frac{k\hat{\sigma}}{\sigma_1/\sqrt{n}} - \delta\right)\right] \\ &= E\left[\Phi\left(\frac{h\left(S_1^2, S_2^2\right)}{\sigma_1/\sqrt{n}} - \delta\right)\right], \end{aligned} \tag{33}$$

where as above

$$\delta = K_\beta\sqrt{\frac{n(R+1)}{JR+1}} = \frac{K_\beta\sigma}{\sigma_1/\sqrt{n}}. \tag{34}$$

The problem is to determine a function $h(S_1^2, S_2^2)$ so that (33) is approximately satisfied for all σ_1^2 and σ_2^2. If tolerance limits on the median are desired, then $\delta = 0$ and the results of Welch (1947) and Aspin (1948) may be used directly. If δ is not zero, the idea behind the Welch-Aspin derivation may still be applied, although the algebra is considerably messier.

The Welch-Aspin approach makes use of certain differential operators in developing an 'asymptotic' series for h. The same approach may be used here, but for first order calculations the additional formalism is not justified in terms of algebraic simplifications. For this reason, the discussion below consists of a straightforward Taylor series derivation. Of course, both methods must give the same answer, and this has been used to provide a check on the calculations.

Begin by rewriting (33) as

$$E\left[\Phi\left(K_\gamma + U\right)\right] = \gamma, \tag{35}$$

where

$$K_\gamma + U = \frac{h\left(S_1^2, S_2^2\right)}{\sigma_1/\sqrt{n}} - \delta. \tag{36}$$

Expand h in a series of inverse powers of n_i,

$$h = h_0 + h_1 + O(1/m^2), \tag{37}$$

where

$$m \equiv \min(n_1, n_2).$$

Up to terms of second order in h we have that

$$U = \frac{h_1(S_1^2, S_2^2)}{\sigma_1/\sqrt{n}} + \frac{h_0(S_1^2, S_2^2)}{\sigma_1/\sqrt{n}} - \frac{K_\beta \sigma}{\sigma_1/\sqrt{n}} - K_\gamma, \tag{38}$$

where we have substituted (34) for δ.

For the zeroth order approximation we approximate $h(S_1^2, S_2^2)$ by

$$h_0(S_1^2, S_2^2) \approx h_0(\sigma_1^2, \sigma_2^2)$$

and we have, $U = 0$ and

$$K_\gamma = \frac{h_0(\sigma_1^2, \sigma_2^2) - K_\beta \sigma}{\sigma_1/\sqrt{n}} \tag{39}$$

or

$$h_0(S_1^2, S_2^2) = \frac{K_\gamma S_1}{\sqrt{n}} + K_\beta S. \tag{40}$$

For the first order expression, we approximate h by

$$h_0(S_1^2, S_2^2) + h_1(S_1^2, S_2^2) \approx h_0(S_1^2, S_2^2) + h_1(\sigma_1^2, \sigma_2^2) =$$
$$K_\beta \sigma \left[1 + \left(\frac{S}{\sigma} - 1\right)\right] + \frac{K_\gamma \sigma_1}{\sqrt{n}} \left[1 + \left(\frac{S_1}{\sigma_1} - 1\right)\right] + h_1(\sigma_1^2, \sigma_2^2) \quad (41)$$

then

$$U = K_\gamma U_1 + K_\beta U_2 + \frac{h_1(\sigma_1^2, \sigma_2^2)}{\sigma_1/\sqrt{n}}, \quad (42)$$

where

$$U_1 = \frac{S_1}{\sigma_1} - 1 \quad (43)$$

and

$$U_2 = \frac{\sigma}{\sigma_1/\sqrt{n}} \left(\frac{S}{\sigma} - 1\right). \quad (44)$$

Let χ_f^2 denote a χ^2 random variable with f degrees of freedom and define

$$V_{n_i} \equiv \frac{\chi_{n_i}^2}{n_i} - 1 \quad (45)$$

for $i = 1, 2$. The U_i can be expressed in terms of the V_{n_i} as follows:

$$U_1 = (1 + V_{n_1})^{1/2} - 1, \quad (46)$$
$$U_2 = \frac{\sigma}{\sigma_1/\sqrt{n}} \left[\left(\frac{\sigma_2^2 (J-1)}{\sigma^2 J} (1 + V_{n_2}) + \frac{\sigma_1^2}{\sigma^2 J} (1 + V_{n_1})\right)^{1/2} - 1\right]. \quad (47)$$

After expanding the square roots in (46) and (47) in power series, one can readily obtain approximations to the first two moments of the U_i suitable for first order calculations:

$$E(U_1) \approx -\frac{1}{4n_1}, \quad (48)$$
$$E(U_1^2) \approx \frac{1}{2n_1}, \quad (49)$$
$$E(U_2) \approx -\frac{1}{4}\left[\frac{a_1}{n_1} + \frac{a_2}{n_2}\right], \quad (50)$$
$$E(U_2^2) \approx \frac{\sigma}{2\sigma_1/\sqrt{n}}\left[\frac{a_1}{n_1} + \frac{a_2}{n_2}\right], \quad (51)$$

and

$$E(U_1U_2) \approx \frac{\sigma_1}{J\sigma/\sqrt{n}}\frac{1}{2n_1}, \tag{52}$$

where

$$a_1 \equiv \frac{\sigma_1^3}{\sigma^3}\frac{n^{1/2}}{J^2} \tag{53}$$

and

$$a_2 \equiv \frac{\sigma_2^4}{\sigma_1\sigma^3}(1-J^{-1})^2 n^{1/2}. \tag{54}$$

The next step is to expand the normal cdf about K_γ, so that (35) may be replaced by the following approximation:

$$\begin{aligned} E\left[\Phi\left(K_\gamma + U\right)\right] = \gamma \approx \Phi(K_\gamma) \\ +\phi(K_\gamma)E(U) - K_\gamma\phi(K_\gamma)E(U^2)/2, \end{aligned} \tag{55}$$

where $\phi(\cdot)$ denotes the standard normal density. The expectation of U can be determined immediately from (42), (48) amd (50). Since

$$E(U^2) = K_\gamma^2 E(U_1^2) + K_\beta^2 E(U_2^2) + 2K_\beta K_\gamma E(U_1U_2) + O(1/m^2), \tag{56}$$

we need only substitute (49), (51) and (52) into (56) in order to complete the evaluation of (55).

To complete these calculations, solve (55) for $h_1(\sigma_1^2, \sigma_2^2)$ (note that h_1 appears through $E(U)$), replace each occurrence of σ_i^2 or σ^2 with S_i^2 or S^2 respectively ($i = 1, 2$) and divide $h_1(S_1^2, S_2^2)$ by S to finally obtain the tolerance limit factor k. The terms of k may then be rearranged to reveal their structure. The following expression for k is one possibility:

$$\begin{aligned} k = K_\beta + \frac{K_\gamma W}{\sqrt{I}} + \frac{W}{4\sqrt{I}}\Bigg[\frac{K_\gamma(K_\gamma^2+1)}{n_1} \\ + \frac{2K_\beta K_\gamma^2\sqrt{I}W}{n_1} + \frac{K_\beta^2 K_\gamma I W^2}{n_1} \\ + \frac{K_\beta\sqrt{I}W^3}{n_1} + \frac{K_\beta^2 K_\gamma I(J-1)^2W^2}{n_2Q^2} \\ + \frac{K_\beta(J-1)^2\sqrt{I}W^3}{n_2Q^2}\Bigg], \end{aligned} \tag{57}$$

where

$$W \equiv (1 + (J-1)/Q)^{-1/2} \tag{58}$$

and

$$Q \equiv \frac{S_1^2}{S_2^2}. \tag{59}$$

The coverage probability for the above approximation as a function of the population variance ratio is plotted in Figure 2 for a (.90, .95) tolerance limit and $J = 5$. Note that for many batches the series solution performs well, though for few batches it is anticonservative.

6 An Alternative Solution for Unknown R

For small samples, the first order approximation developed above may not be adequate, and higher order calculations are clearly prohibitive. An alternative approach is to view the problem as an integral equation, following Trickett and Welch (1954).

If one defines

$$\tau \equiv JR + 1 \tag{60}$$

then (24) may be written as

$$E_\nu\left[T_{n_1+n_2}\left(k(\tau)(n_1+n_2)^{1/2}\sqrt{\frac{XI}{I-1}+\frac{1-X}{\tau}}, \delta(\tau)\right)\right] = \gamma, \tag{61}$$

where

$$\delta = \sqrt{n}K_\beta B = K_\beta\sqrt{I\left(1+\frac{J-1}{\tau}\right)} \tag{62}$$

and B is defined in (14). The parameter τ may be estimated by the mean square ratio (59):

$$Q = \tau F_{n_1,n_2}, \tag{63}$$

where F_{n_1,n_2} denotes a random variable having an F distribution with n_1 and n_2 degrees of freedom. The tolerance limit problem reduces to determining a function $\tilde{k}(Q)$ such that

$$\begin{aligned} \gamma &= P\left(\frac{Z+\delta(\tau)}{\sqrt{\frac{IY_1}{I-1}+\frac{Y_2}{\tau}}} \leq \tilde{k}(Q)\right) \\ &= P\left[Z \leq \tilde{k}\left(\frac{\tau n_2 Y_1}{n_1 Y_2}\right)\sqrt{\frac{IY_1}{I-1}+\frac{Y_2}{\tau}} - \delta(\tau)\right] \end{aligned} \tag{64}$$

for all values of $\tau \geq 1$, where Z, Y_1, and Y_2 are as in Section 3. This is equivalent to the integral equation

$$V_\tau(\tilde{k}) \equiv E_\nu \left[T_{n_1+n_2} \left(\tilde{k}(Q)(n_1+n_2)^{1/2} \sqrt{\frac{XI}{I-1} + \frac{1-X}{\tau}}, \delta(\tau) \right) \right] = \gamma, \tag{65}$$

where

$$Q = \frac{\tau n_2 X}{n_1(1-X)} \tag{66}$$

and the expectation is with respect to a beta density with parameter $\nu = (n_1/2, n_2/2)$.

In Section 5, we derived an approximation to $\tilde{k}(Q)$ which we label here $\tilde{k}_0(Q)$. We intend to improve this approximation by replacing it by $\tilde{k}_1(Q) = \tilde{k}_0(Q) + \psi(Q)$ and we will employ

$$\underline{k}(\epsilon, Q) \equiv \tilde{k}_0(Q) + \epsilon\psi(Q), \tag{67}$$

noting that $\tilde{k}_1(Q) = \underline{k}(1, Q)$.

Expanding $V_\tau\left[\underline{k}(\epsilon, Q)\right]$ in a Taylor series gives the first order approximation

$$\gamma(\epsilon) = V_\tau\left[\tilde{k}_0(Q) + \epsilon\psi(Q)\right] \approx V_\tau(\tilde{k}_0(Q)) + \epsilon \left.\frac{dV_\tau}{d\epsilon}\right|_{\epsilon=0}. \tag{68}$$

Employing this result, the equation leading to our next approximation may be written as

$$\begin{aligned} \gamma = \gamma(1) = \\ & V_\tau(\tilde{k}_0) + E_\nu \left[\psi(Q)(n_1+n_2)^{1/2} \sqrt{\frac{XI}{I-1} + \frac{1-X}{\tau}} \right. \\ & \left. \cdot t_{n_1+n_2} \left(\tilde{k}_0(Q)(n_1+n_2)^{1/2} \sqrt{\frac{XI}{I-1} + \frac{1-X}{\tau}}, \delta \right) \right], \end{aligned} \tag{69}$$

where $t_{n_1+n_2}(\cdot,\cdot)$ denotes the noncentral t density. The noncentral t density with f degrees of freedom and noncentrality parameter δ may be calculated by means of the following formula (Odeh and Owen, 1980, p. 272):

$$t_f(x, \delta) = \frac{f}{x} \left[T_{f+2} \left(x\sqrt{\frac{f+2}{f}}, \delta \right) - T_f(x, \delta) \right]. \tag{70}$$

Since there are computer subroutines available for determining the non-central t cdf (see, e.g., Griffiths and Hill, 1985), (70) is very useful for computation.

The first term on the right hand side of (69), $V_\tau(\tilde{k}_0)$, may be evaluated numerically for given τ since $\tilde{k}_0(Q)$ is a known function.

The second term can not be evaluated without knowing ψ. To simplify matters we shall pretend that $\psi(Q)$ assumes a constant value and can be factored out of the expectation. The Trickett-Welch approach consists of replacing $\psi(Q)$ by $\psi(q_0)$ where q_0 is that value of Q corresponding to the mean x_0 of the beta random variable X, i.e.

$$q_0 \equiv \frac{\tau n_2 x_0}{n_1(1-x_0)} = \frac{\tau n_2 n_1/(n_1+n_2)}{n_1 n_2/(n_1+n_2)} = \tau. \tag{71}$$

Thus we have

$$\gamma \approx V_\tau(\tilde{k}_0) + \psi(\tau) V_{1\tau}(\tilde{k}_0) \tag{72}$$

where

$$V_{1\tau}(\tilde{k}_0) \equiv \tag{73}$$
$$E_\nu\left[(n_1+n_2)^{1/2}\sqrt{\frac{XI}{I-1}+\frac{1-X}{\tau}}\right.$$
$$\left.\cdot t_{n_1+n_2}\left(\tilde{k}_0(Q)(n_1+n_2)^{1/2}\sqrt{\frac{XI}{I-1}+\frac{1-X}{\tau}},\delta\right)\right],$$

$$\psi(\tau) \approx \frac{\gamma - V_\tau(\tilde{k}_0)}{V_{1\tau}(\tilde{k}_0)}, \tag{74}$$

and

$$\tilde{k}_1(\tau) = \tilde{k}_0(\tau) + \psi(\tau). \tag{75}$$

Our numerical integration depends on the use of $\tilde{k}_0$ for a mesh of Q or τ values from 0 to ∞. We can compute $\tilde{k}_1$ for the same mesh. This $\tilde{k}_1(Q)$ can be used as the $\tilde{k}_0(Q)$ for the next iteration.

The approximation which allows $\psi(Q)$ to be removed from the integrand is crude. It is certainly not obvious that this procedure will provide any improvement on the first approximation. In fact, if one were to implement the algorithm presented in this section on a computer, one would see that the coverage probability improves only slightly before the successive approximations begin to diverge. By employing this method as improved in the following section, we can do much better. The improvements of Section 7 result in an algorithm which provides a solution to the tolerance limit problem of unknown R that is (for practical purposes) exact. All mention of results of applying the Trickett-Welch method in this paper refer to the modification to be discussed in the next section.

7 A Modification of the Trickett-Welch Approach

In order to obtain useful results from the integral equation approach of Section 6, it is necessary to improve the rough approximation by which the unknown function ψ is removed from the the expectation in (69). The technique by which this approximation is improved is based on evaluating $\psi(Q)$ for a value q_1 of Q corresponding to x_1 of X where the integrand of $V_{1\tau}$ in (73) achieves its maximum, instead of x_0, the mean of the beta density, which may be close to where this density achieves its maximum.

For any value of τ, we can determine the desired desired peak $x_1(\tau)$ numerically, and define

$$q_1 \equiv \frac{\tau n_2 x_1(\tau)}{n_1(1 - x_1(\tau))}. \tag{76}$$

It is fortunate that in our tolerance limit problem, the value $x_1(\tau)$ is nearly independent of τ. Thus, $\psi(Q)$ can be evaluated at or very nearly at a specified grid of q_1 values by adjusting τ after the nearly constant value x_1^* of $x_1(\tau)$ is approximated for a typical τ value.

One difficulty with the above proposal arises from the fact that, strictly speaking, τ should only be taken to be greater than one, in which case the range of q_1 values is from $n_2 x_1^*/[n_1(1 - x_1^*)]$ to ∞ instead of from 0 to ∞ as is required for the numerical integration. Since $n_2 x_1^*/[n_1(1 - x_1^*)]$ turns out to be relatively small, we translate the value of q_1 by this amount, so that the range of q values will be 0 to ∞. In other words, we replace $\tilde{k}_0(q_1)$ in the approximation

$$\gamma = \gamma(1) = V_\tau(\tilde{k}_0(q_1)) + \psi(q_1) V_{1\tau}(\tilde{k}_0(q_1)) \tag{77}$$

by $\tilde{k}_0\{q_1 - n_2 x_1^*/[n_1(1 - x_1^*)]\}$. After this approximation is carried out, the method can be iterated using

$$\tilde{k}_1\left[q_1 - \frac{n_2 x_1^*}{n_1(1 - x_1^*)}\right] = \tilde{k}_0\left[q_1 - \frac{n_2 x_1^*}{n_1(1 - x_1^*)}\right] + \psi(q_1) \tag{78}$$

to replace $\tilde{k}_0$.

With each iteration the value of the constant x_1^* is likely to change and should be recalculated.

The above simple improvement of the integral mean value theorem approximation underlying the Trickett-Welch approach enables one to calculate tolerance limit factors which provide very nearly the nominal coverage probability even for few batches and small batch size.

The Trickett-Welch approach, possibly with modifications similar to those discussed in this section, promises to be applicable to other problems, two of which are considered below.

8 Other Applications of the Trickett-Welch Approach

The Trickett-Welch approach can be applied to a wide range of problems of inference in the presence of a nuisance parameter. Two examples of such problems are outlined in this section.

8.1 Confidence Intervals for the Population Mean

For the random effects model of Section 1 a two sided confidence interval for the population mean is desired which attains nearly the nominal confidence level whatever the population gintraclass correlation

$$\rho \equiv \frac{\sigma_b^2}{\sigma_b^2 + \sigma_e^2} = \frac{R}{R+1}. \tag{79}$$

Let $D_1(\cdot)$ be an unspecified function of the mean square ratio (59). With notation as in Sections 5 and 6 we have

$$\gamma = P(\hat{\mu} - D_1 S \leq \mu \leq \hat{\mu} + D_1 S). \tag{80}$$

This is easily shown to be equivalent to the integral equation

$$\frac{1+\gamma}{2} = E\left[\Phi\left(\frac{D_1(S_1^2/S_2^2)S}{\sigma_{\hat{\mu}}}\right)\right], \tag{81}$$

where the expectation is with respect to the distributions of the mean squares. The methodology presented in this paper can be applied directly. An important feature of this example is that it provides a method which, for a particular simple situation, avoids the problematic issue of 'when to pool'.

8.2 Testing the Equality of Two Normal Percentiles

Another thinly disguised version of the Behrens-Fisher problem is the problem of testing the equality of two normal percentiles where population means and variances are unknown. Two statistics for performing such a test are proposed by Cox and Jaber (1985). These tests require simulation in order to obtain approximate critical values for the test statistics. The method outlined below, though its properties have yet to be examined, requires no Monte-Carlo tables.

We wish to test equality of the $100\beta th$ percentiles of two normal populations on the basis of simple random samples from each population. That is, we are interested in testing the null hypothesis

$$H_0 : \mu_1 + K_\beta\sigma_1 = \mu_2 + K_\beta\sigma_2 \tag{82}$$

against the alternative

$$H_1 : \mu_1 + K_\beta\sigma_1 \neq \mu_2 + K_\beta\sigma_2, \tag{83}$$

where μ_i and σ_i are the population mean and standard deviantions for $i = 1,2$ and K_β is defined in (7).

Denote the sample means and variances by $\bar{X}_i$ and S_i^2 and let the sample sizes be n_i. Define the statistic

$$T \equiv (\bar{X}_1 + K_\beta S_1) - (\bar{X}_2 + K_\beta S_2) \tag{84}$$

We propose to reject the null hypothesis when $|T|$ is sufficiently large. A function D_2 which provides such a test (of size $1 - \gamma$) can be shown to satisfy the following integral equation, where as above the expectations are with respect to the mean squares:

$$E\left[\Phi\left(\frac{D_2 - K_\beta(S_1 - S_2 + \sigma_2 - \sigma_1)}{\sqrt{\sigma_1^2/n_1 + \sigma_2^2/n_2}}\right)\right] - \tag{85}$$

$$E\left[\Phi\left(\frac{-D_2 - K_\beta(S_1 - S_2 + \sigma_2 - \sigma_1)}{\sqrt{\sigma_1^2/n_1 + \sigma_2^2/n_2}}\right)\right] = \gamma$$

which is in a form to which the techniques of this paper may be applied.

9 The Distributions of the Tolerance Limits

Once the function k has been determined it is straightforward to calculate the cumulative distribution function of the tolerance limit. It is obviously preferable to compare distributions of confidence bounds rather than merely confidence levels, and we make such a comparison in this section.

We consider a (.90,.95) lower tolerance limit for a normal population with tenth percentile zero and variance one. In Figure 3, the cumulative distributions for $I = J = 5$ of the proposed tolerance limit are presented for various values of the intraclass correlation $\rho = R/(R+1)$.

Note that all of the curves pass very nearly through (0,.95), indicating the striking success that we have had at removing the nusiance parameter, even for as few as five batches. As the intraclass correlation is increased the random effects sample goes from behaving essentially like a single sample of size $n = IJ$ when $\rho = 0$ to being equivalent to a single batch of size I when $\rho = 1$.

In Figure 4 three cdfs are plotted, corresponding to the Mee-Owen method, the proposed method and the the solution for known R. The intraclass correlation is taken to equal zero and the sample size is again $I = J = 5$. Note that the result based on the Trickett-Welch approach is clearly preferable to the Mee-Owen solution and doesn't fare too badly when compared to the known-R solution.

10 Discussion

The situation of primary interest to the aircraft industry, (.90,.95) lower tolerance limits, is used here for illustration. Four methods have been presented in this paper: the Mee-Owen method (Section 2), a modified Mee-Owen method (Section 2), a method based on the Welch-Aspin series (Section 5), and a method based on an integral equation (Sections 6, 7). The coverage probability functions corresponding to these methods are numbered 1-4 in Figure 5 for five batches each of size five.

The integral equation approach virtually removes the nuisance parameter from the problem. The Mee-Owen method has the disadvantage of being substantially conservative when the variance ratio is small.

Only a slight reduction in this conservatism has resulted from the modification of the confidence level of the variance ratio estimate (Section 2, Table 2).

The Welch-Aspin series solution is clearly not suitable for as few as five batches, as discussed in Section 5. However, it is easy to compute and provides an adequate starting function for the iterative integral equation approximation (69).

From the rescaled plot of the coverage probability function for the integral equation solution (Figure 6) it can be seen that for $R > 1$ the actual coverage probability differs from .95 by no more than ±.00005. This small difference can be attributed to the limited accuracy of the numerical integration. For $R < 1$, however, the difference in the actual and nominal coverage probability increases substantially, but never does it reach a magnitude that warrants concern for applications.

Figure 6 illustrates the convergence of the Trickett-Welch approach for various values of the intraclass correlation. Note that for practical purposes ten iterations is adequate, although some slight improvement may result from considering more iterations.

11 Example

The data in Table 3 are a pseudo-random sample of 25 from a normal distribution with mean 50 and standard deviation 10. These data have been arbitrarily grouped into five batches of five. By fitting a one-way random effects model to these data one obtains :

$$MS_b = 89.88, \quad MS_e = 158.6, \tag{86}$$

$$\hat{\mu} = 48.30, \quad \hat{\sigma}_X^2 = 144.9. \tag{87}$$

A $(.90, .95)$ lower tolerance limit is of the form

$$T = \hat{\mu} - K\hat{\sigma}_X. \tag{88}$$

For the method of Mee and Owen (1983) $K = 1.90$. If the Mee-Owen method is modified as suggested in in Section 2, then K only decreases to 1.89. The series solution of Section 5 gives $K = 1.78$ and the integral equation of Section 6 results in $K = 1.83$. The tolerance limit estimates are, respectively, 25.42, 25.54, 26.82 and 26.29. These values may be compared with the tolerance limit estimate for the pooled data, which is 26.00.

12 Conclusion

One-sided tolerance limits for random effects models is a topic of considerable importance in engineering statistics. The purpose of this paper has been to consider this tolerance limit problem from the point of view of the Welch interpretation of the Behrens-Fisher problem. This approach leads to a method which provides essentially the nominal coverage probability whatever the value of the nuisance parameter. We have demonstrated that in addition to excellent coverage properties, the distribution of the proposed tolerance limit compares favorably with an existing procedure and with an exact solution for known nuisance parameter.

References

[1] Aspin, A. A. (1948), "An Examination and Further Development of a Formula Arising in the Problem of Comparing Two Mean Values", *Biometrika* 35, 88-96.

[2] Cox, T. F. and Jaber, K. (1985) "Testing the Equality of Two Normal Percentiles", *Communications in Statistics- Simulation and Computation*, 14, 345-356.

[3] Fleiss, J. L. (1971) "On the Distribution of a Linear Combination of Independent Chi Squares", *Journal of the American Statistical Association*, 66, 142-144.

[4] Griffiths, P. and Hill, I. D. (1985) *Applied Statistics Algorithms*, Wiley, New York.

[5] Lemon, G. H. (1977) "Factors for One-Sided Tolerance Limits for Balanced One-Way ANOVA Random-Effects Model", *Journal of the American Statistical Association*, 72, 676-680.

[6] Linnik, Y. V. (1968), *Statistical Problems with Nuisance Parameters*, Translations of Mathematical Monographs Volume 20, American Mathematical Society, Providence, RI.

[7] Mee, R. W. and Owen, D. B. (1983) "Improved Factors for One-Sided Tolerance Limits for Balanced One-Way ANOVA Random Model", *Journal of the American Statistical Association*, 78, 901-905.

[8] Mil Handbook 17B (1988), *Polymer Matrix Composites, Volume I: Guidelines*, Naval Publications and Forms Center, Philadelphia.

[9] Neal, D., Vangel, M. and Todt, F. (1987), "Determination of Statistically Based Composite Material Properties", in *Engineered Materials Handbook, Vol. 1, Composites*, ed. Cyril A. Dostal, American Society of Metals Press, Metals Park, OH.

[10] Odeh, R. E. and Owen, D. B. (1980), *Tables of Normal Tolerance Limits, Sampling Plans and Screening*, Marcel Dekker.

[11] Owen, D. B. (1968), "A Survey of Properties and Applications of the Noncentral t-Distribution", *Technometrics*, 10, 445-478.

[12] Ray, W. D. and Pitman A. E. N. T. (1961), "An Exact Distribution of the Fisher-Behrens-Welch Statistic for Testing the Difference Between the Means of Two Normal Populations with Unknown Variances", *Journal of the Royal Statistical Society*, 23, 377-84.

[13] Satterthwaite, F. E. (1946), "An Approximate Distribution of Estimates of Variance Components", *Biometrics Bulletin*, 2, 110-114.

[14] Searle, S. R. (1971), *Linear Models*, John Wiley and Sons, N.Y.

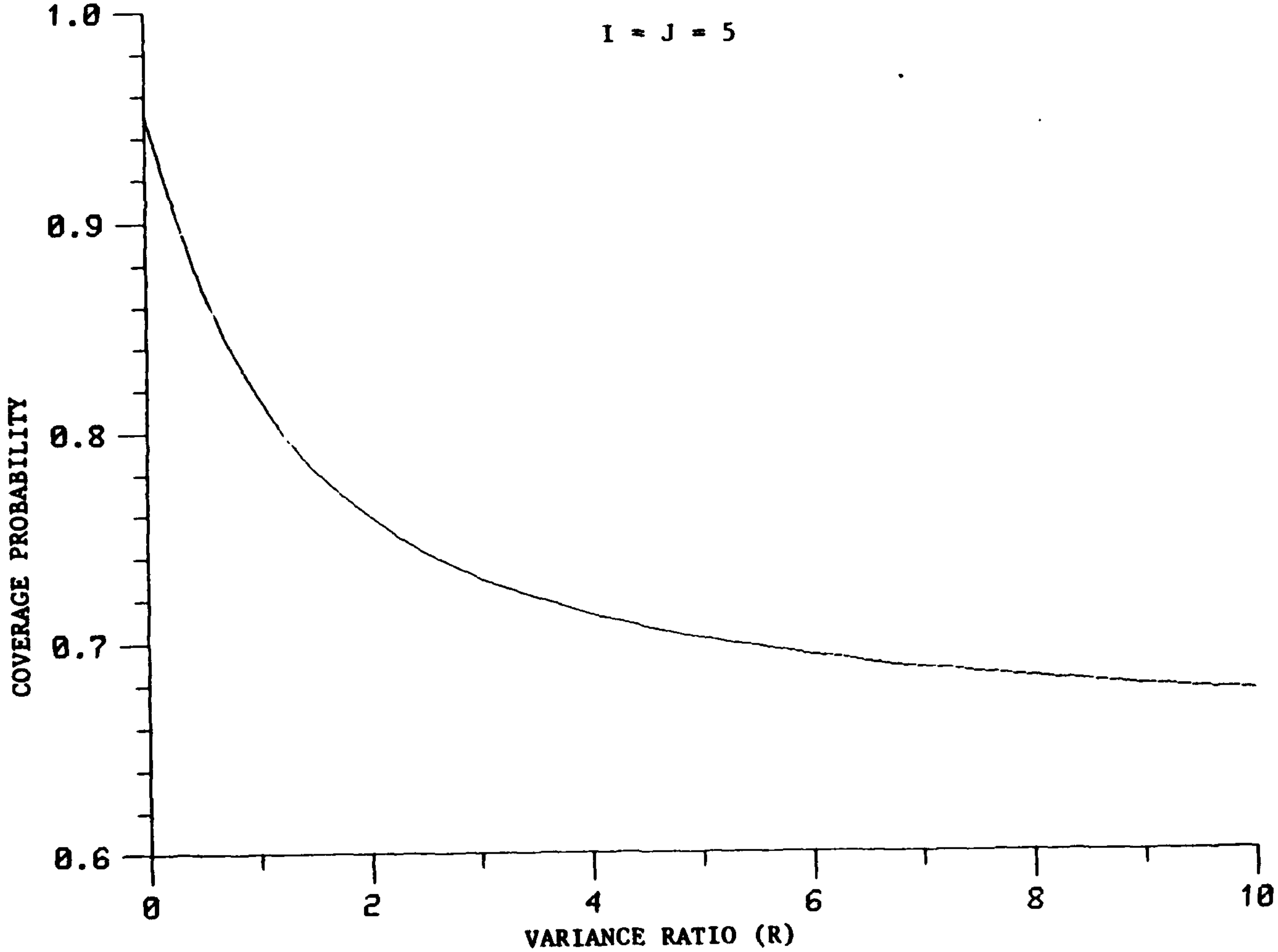
[15] Trickett, W. H. and Welch, B. L. (1954), "On the Comparison of Two Means: Further Discussion of Iterative Methods for Calculating Tables", *Biometrika*, 41, 361-374.

[16] Welch, B. L. (1947), "The Generalization of Student's Problem When Several Different Population Variances are Involved", *Biometrika*, 34, 28-35.

Figure 1

Coverage probability for a (.90, .95) lower tolerance limit using the pooled data method as a function of the variance ratio.

I = J = 5



<u>Figure 2</u>

Coverage probabilities for (.90, .95) tolerance limits using the first order Welch series method.

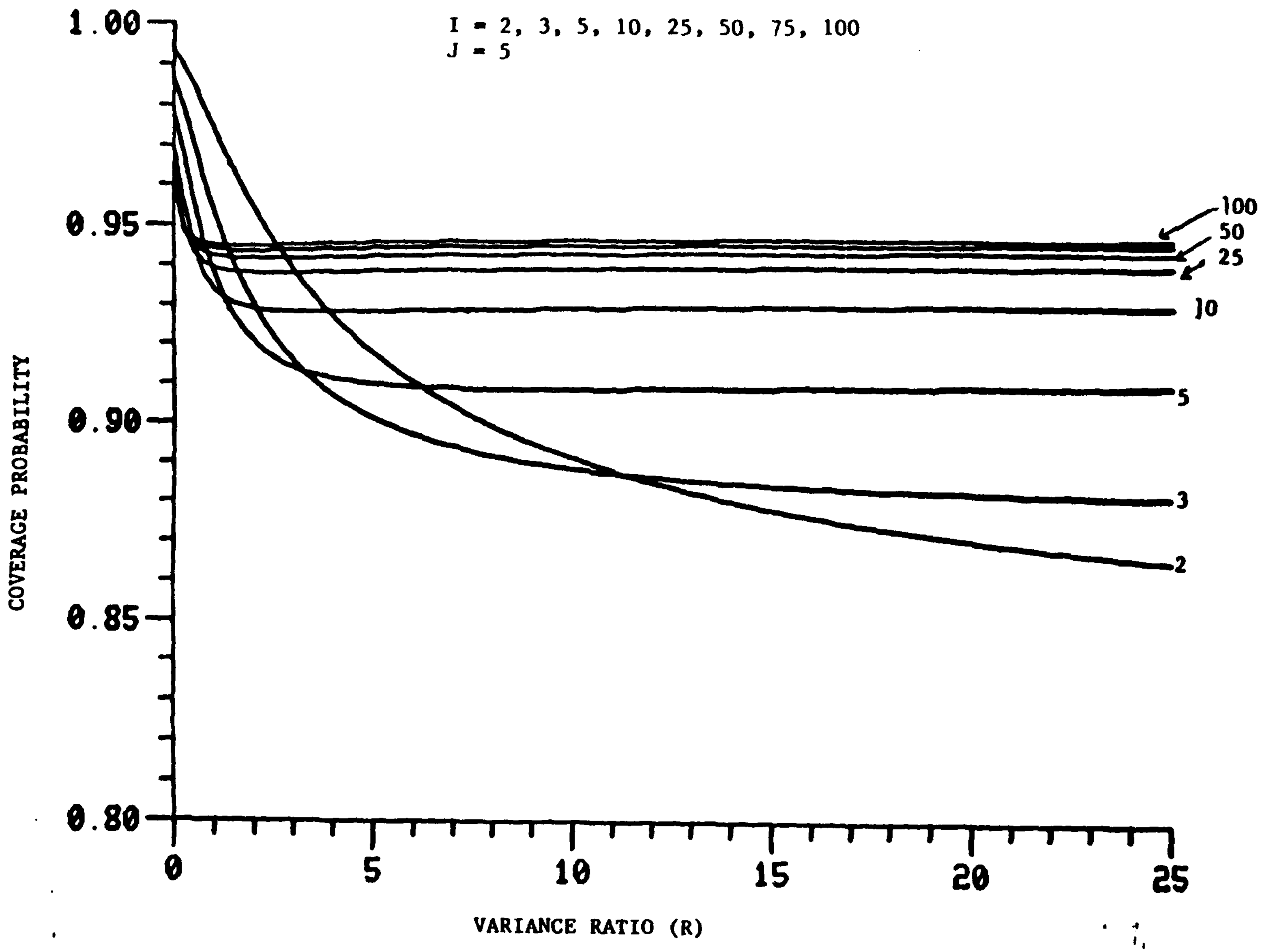
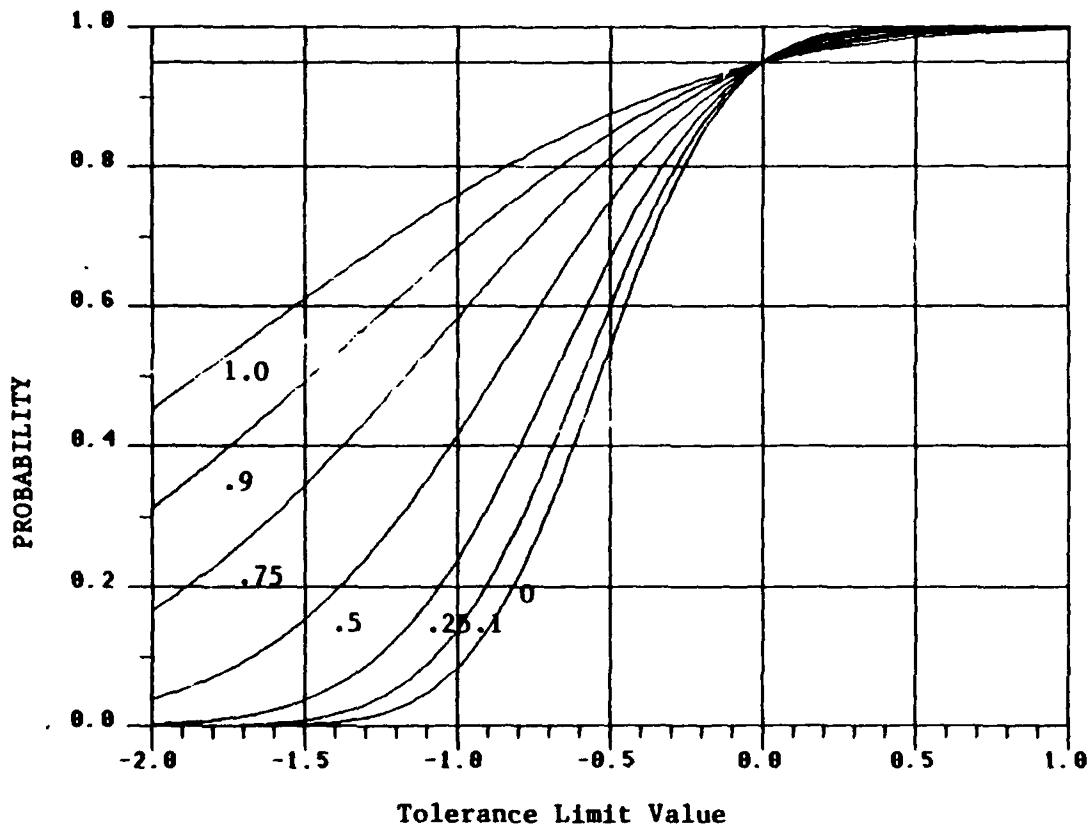


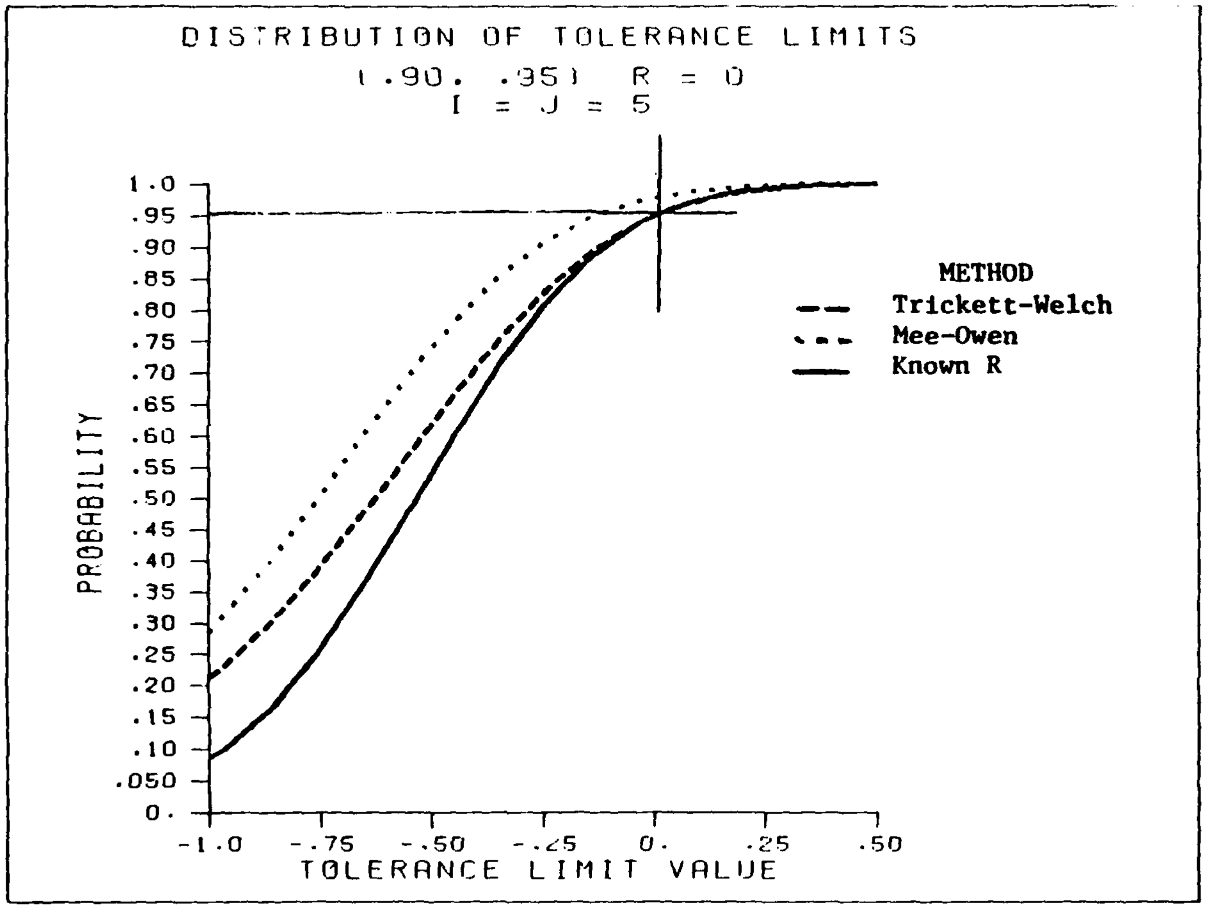
Figure 3

Distributions of (.90, .95) lower tolerance limits obtained by the Trickett-Welch method for various values of the intraclass correlation.

I = J = 5



$$\rho \equiv \frac{\sigma_b^2}{\sigma_b^2 + \sigma_e^2} = \frac{R}{R+1}$$

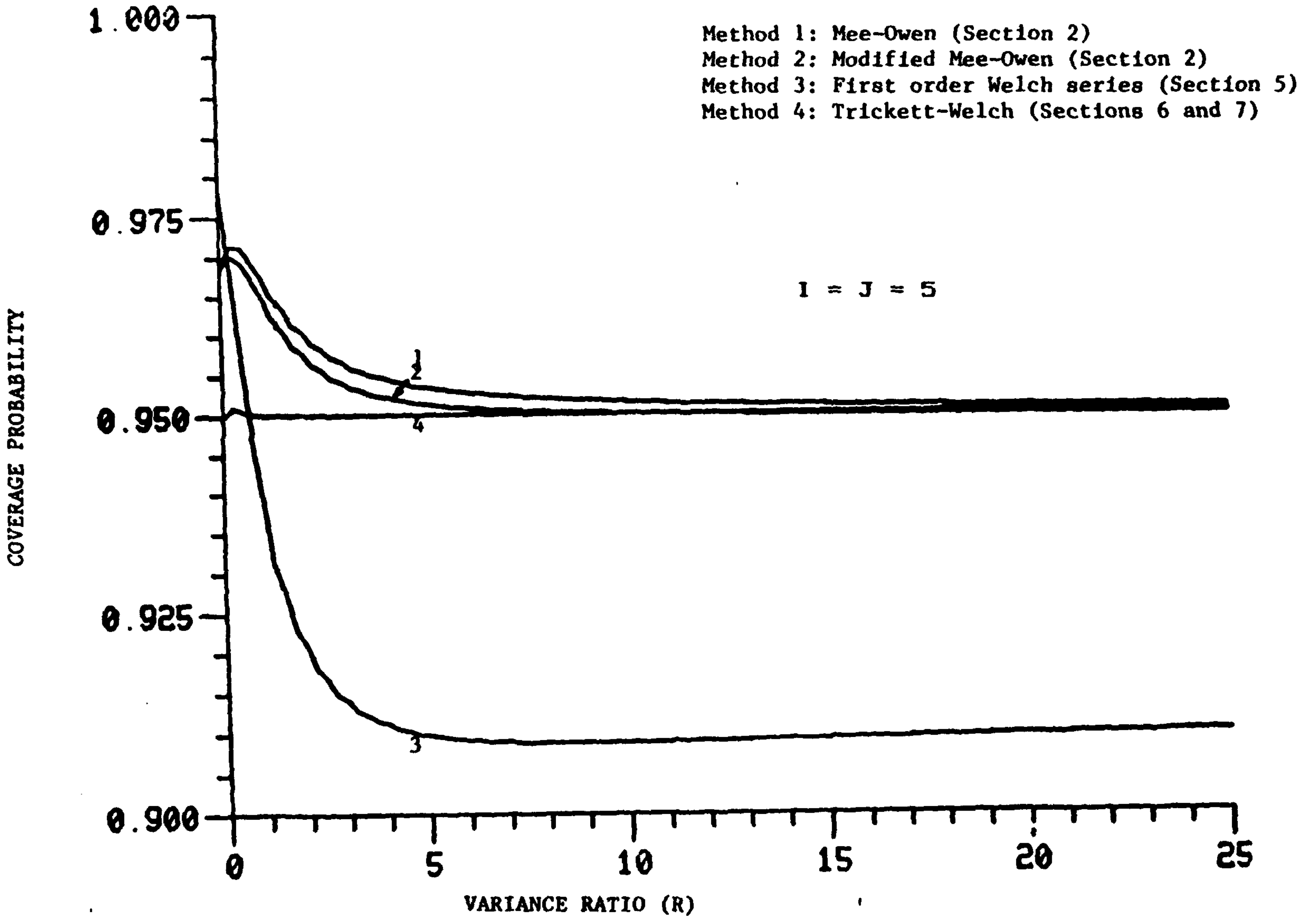


DISTRIBUTION OF TOLERANCE LIMITS
(.90, .95) R = 0
I = J = 5
METHOD
Trickett-Welch
Mee-Owen
Known R
PROBABILITY
1.0
.95
.90
.85
.80
.75
.70
.65
.60
.55
.50
.45
.40
.35
.30
.25
.20
.15
.10
.050
0.
-1.0
-.75
-.50
-.25
0.
.25
.50
TOLERANCE LIMIT VALUE

Figure 4

Figure 5

Coverage probabilities of (.90, .95) tolerance limits calculated by four methods as functions of the variance ratio.



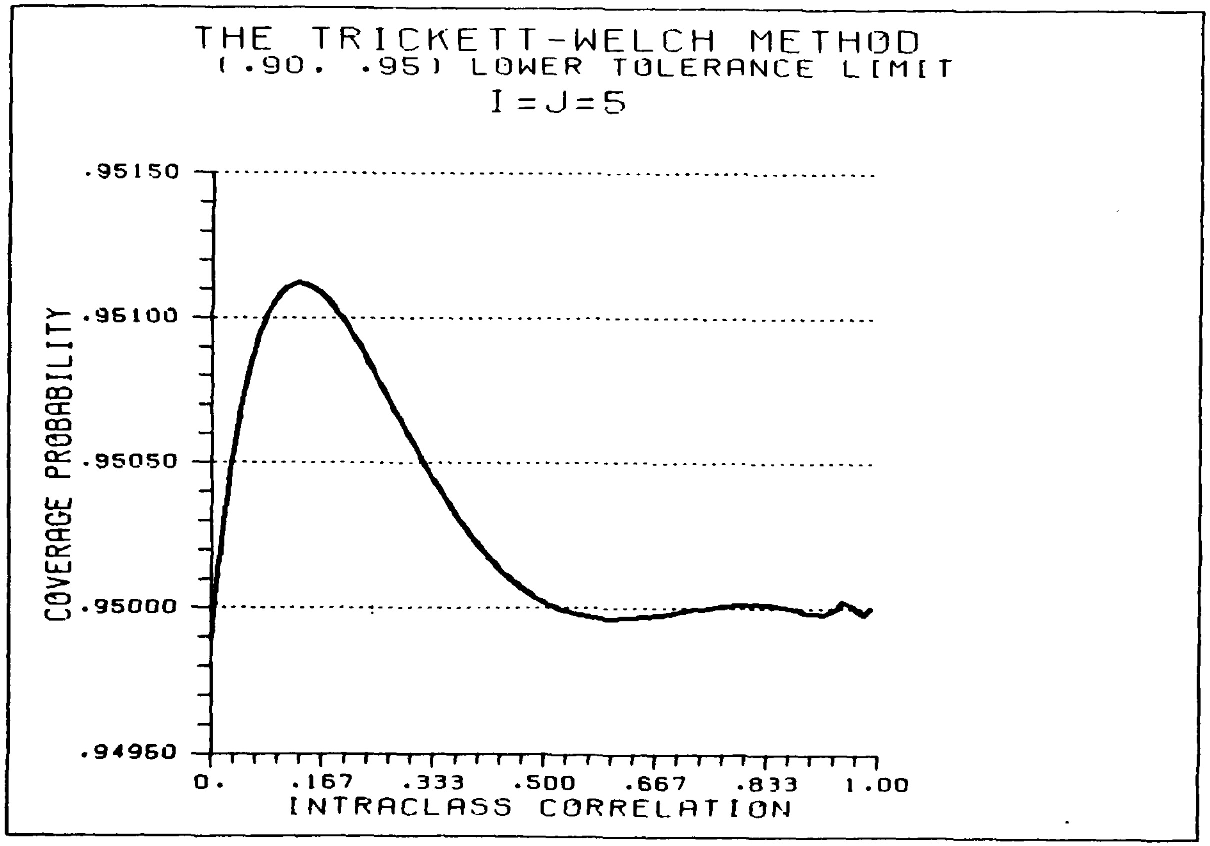
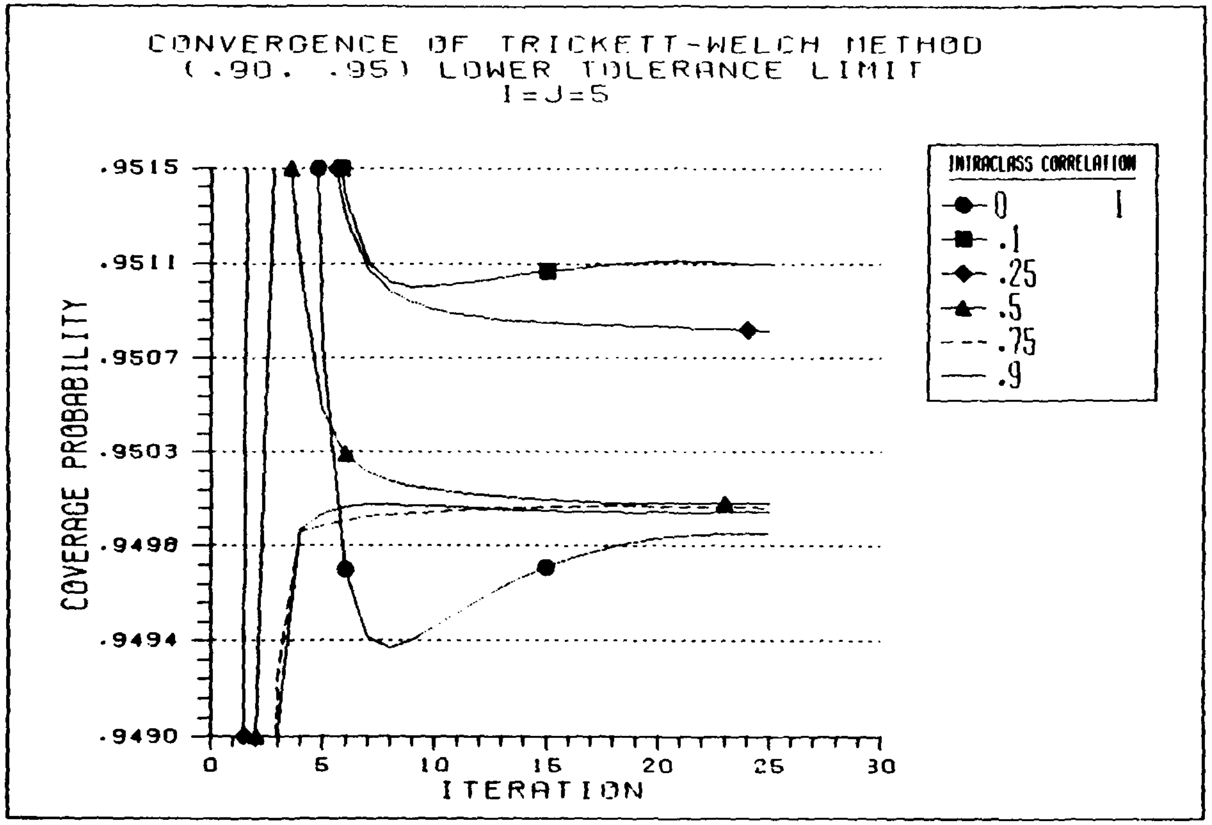

THE TRICKETT-WELCH METHOD
(.90. .95) LOWER TOLERANCE LIMIT
I=J=5
COVERAGE PROBABILITY
.95150
.95100
.95050
.95000
.94950
0.
.167
.333
.500
.667
.833
1.00
INTRACLASS CORRELATION

Figure 6



CONVERGENCE OF TRICKETT-WELCH METHOD
(.90, .95) LOWER TOLERANCE LIMIT
I=J=5
COVERAGE PROBABILITY
.9515
.9511
.9507
.9503
.9498
.9494
.9490
0
5
10
15
20
25
30
ITERATION
INTRACLASS CORRELATION
0
.1
.25
.5
.75
.9

Figure 7

Table 1

η Values for (β, γ) Tolerance Limits (Mee and Owen, 1983, p.90)

		γ = .90	γ = .95	γ = .99
	.90	.78	.85	.94
β	.95	.79	.86	.95
	.99	.81	.875	.96

Table 2

η Values for (.90, .95) Tolerance Limits for the Mee-Owen Method.

ROWS: Number of batches

COLUMNS: Batch size

	3	4	5	6	7	8	9	10
3	.63	.69	.73	.75	.76	.77	.78	.79
4	.75	.78	.80	.81	.82	.82	.83	.83
5	.80	.82	.83	.83	.83	.84	.84	.84
6	.82	.83	.83	.84	.84	.84	.84	.84
7	.82	.83	.83	.84	.84	.84	.84	.84
8	.82	.83	.83	.84	.84	.84	.84	.84
9	.82	.83	.83	.84	.84	.84	.84	.84
10	.82	.83	.83	.83	.84	.84	.84	.84

Table 3
Example Data

Batch 1	Batch 2	Batch 3	Batch 4	Batch 5
59.45	38.46	30.58	55.65	60.41
40.70	43.24	29.15	50.68	64.45
24.67	66.82	46.29	67.62	36.57
30.60	51.95	63.85	42.02	59.76
52.51	38.50	51.71	41.09	40.84

Appendix
FORTRAN source code listings

Listed below are the routines used to perform the calculations in this paper. All of the required software is listed with the following exceptions:

1) Routines in the IMSL library
2) TEKTRONIX PLOT-10 graphics subroutines
3) The noncentral-t distribution with non-integer degrees of freedom ('TNC', Algorithm AS 243, Applied Statistics (1989) v. 38)
4) Routines called by 'TNC' above, all of which are in Griffiths and Hill (1985).

The routines listed in this appendix are available in computer readable form at no charge from the author. Send a floppy disk (IBM-PC) or a magnetic tape for a copy of the source files.

This code is a prototype intended as a research tool. It is not suitable in the present form for general purpose use.

Main programs:

TRICK -- Program to calculate tolerance limit factor by the modified Trickett-Welch approach for a balanced nested mixed model, a simple generalization of the model considered in this paper.

PLTCDF -- Program to calculate and plot distribution functions for tolerance limits. This program uses as input tolerance limit factor files created by program 'TRICK'.

COVRGE -- Program to calculate the coverage probability vs. intraclass correlation functions from tolerance limit factor files created by program 'TRICK'

Subroutines:

EVCDF -- Routine to evaluate the cdf of a tolerance limit. Called by 'TLMCDF'.

FNCK -- Function called by root finder in 'INVNCT'.

FNCN -- Function called by root finder in 'INVSPN'

FNCR -- Function called by root finder in 'KR'.

FNCS -- Function called by maximization routine in 'FSUP'.

FNCY -- Function called by numerical integration subroutine in 'GENDS2'.

FNCZ -- Function called by numerical integration subroutine in 'EVCDF'.

GENDS2 -- Function to calculate cdf of a generalized noncentral-t random variable.

INISPL -- Subroutine to initialize spline interpolation of tolerance limit factor.

INIT -- Initialization routine for 'TRICK'.

INTEQN -- Top-level subroutine for solving Trickett-welch integral equation. Called by 'TRICK'.

INVNCT -- Subroutine to determine noncentral-t quantiles.

INVSPN -- Subroutine to perform inverse spline interpolation. Called by 'MESH'.

KFACT -- Subroutine to determine tolerance limit factor for a simple random sample from a normal distribution.

KMO -- Subroutine to calculate the Mee-Owen tolerance limit factor. Since the Satterthwaite degrees of freedom need not be an integer, it is because of this routine that the Applied Statistics subroutine 'TNC' is used. For integer degrees of freedom the IMSL routine is adequate.

KR -- Routine to determine the tolerance limit factor for known variance ratio.

KSPLN -- Spline interpolation for tolerance limit factor.

MESH -- Subroutine to improve the mesh of nuisance parameter values. Initially, the Welch series is evaluated for equally spaced values of the ratio of mean squares. A spline is fit to this function, the ordinate is divided into equal intervals, and the spline is inverted to provided new abscissa values which will be closer together where the function has a larger derivative. This new mesh is used for all future iterations.

NCTD1N -- Called by 'NCTDRV'. Noncentral-t density.

NCTD2N -- Called by 'NCTDRV'.

NCTD3N -- Called by 'NCTDRV'.

NCTD4N -- Called by 'NCTDRV'.

NCTD5N -- Called by 'NCTDRV'.

NCTDRV -- Subroutine to recursively calculate any of the first five derivatives of the noncentral t distrivution. Derivatives beyond the first are not used in the paper.

NEXTK -- Subroutine which calculates the next iteration of the modified Trickett-Welch procedure. Called by 'INTEQN', which is called by program 'TRICK'.

SUP -- Function to find the maximum of the Trickett-Welch integrand in order to improve integral mean value approximation as discussed in Section 7. Called by 'NEXTK'.

TLMCDF -- Subroutine to calculate the distribution of the tolerance limit. Called by 'PLTCDF' and 'COVRGE'.

WELCH -- Function to calculate the first order Welch series approximation.

XINT -- Function to evaluate the two integrands for the modified Trickett-Welch algorithm. Called by 'NEXTK' and 'FNCS'.

```
      program covrge
c     ----------------------------------------------------------------
c
c      Mark Vangel, Nov. 1988
c
c         Program to determine points on  coverage probability vs.
c      intraclass correlation curves for tolerance limit factors.
c      This program uses as input files created by 'TRICK'.
c
c      ** Note: This program sends extensive output to the
c               terminal which may be used for later plotting.
c               On many computers this output will have to be
c               redirected to a file.
c
      real          x(500), xk(500), xdx(100), cov(100)
      integer       n(3)
      character*20 flenme, file2
      character*3  citer
      character*1  ans
      common /kw/ known, rho, xknown
      data lfn /10/
c
c     -- Filename (specified in 'TRICK')
      write (*,*) 'filename ?'
      read  (*,'(a20)')  flenme
c
c     -- Number of points at which coverage probability is to
c        be determined
      write (*,*) 'points per curve ?'
      read (*,*) nrho
c
c      -- Indices for tolerance limit factor files.  These files
c         are output from 'TRICK'.  The index corresponds to the
c         iteration, it follows the hyphen in the file's name.
      write (*,*) '1=range of indices, 2=individual indices ?'
      read  (*,*) iopt
      if (iopt .eq. 1) then
         write (*,*) 'min index,  max index ?'
         read  (*,*)  imin     ,  imax
         ndp = imax -imin +1
         do 10 i=1, ndp
            xdx(i) = imin +(i-1)
10      continue
      else
         ndp = 0
20       continue
            write (*,*) 'index (0 to quit) ?'
            read  (*,*)   idx1
            if (idx1 .eq. 0) go to 30
            ndp = ndp +1
            xdx(ndp) = idx1
            go to 20
30       continue
      end if
c
c      -- Get parameters corresponding to tolerance limit factor
c         from files refered to above.
c         (nfix=1 for cased treated in the paper)
      write (*,*) ' nfix, i, j ?'
      read  (*,*)   n
      k = n(2)
      l = n(3)
      write (*,*) ' p ?'
      read  (*,*)   p
```

```
      write (*,*) ' confidence ?'
      read  (*,*)   g
c
c      -- Intraclass correlation is not regarded as known.
      known  = .false.
c
c      -- Loop over tolerance limit facotr input files.
      do 40 i=1, ndp
c
c      -- Build the file's name using the root name 'flenme' and
c         the index number corresponding to the current iteration.
         write (unit=citer, fmt='(a1,i2)') '-',int (xdx(i))
         if (citer (2:2) .eq. ' ') citer (2:2) = '0'
         lstnbk = 20
50       continue
            if (flenme (lstnbk:lstnbk) .eq. ' ') then
               lstnbk = lstnbk -1
               go to 50
            end if
         file2 = flenme (1:lstnbk) //citer
c
c     -- Loop over points on each curve
         dr = 1/ float (nrho -1)
c
c    -- Header for curve (you may want to direct succeeding output
c       to a file for later plotting).
         write (*,*)
         write (*,*) ' Number of fixed factors  : ',n(1)
         write (*,*) ' Number of random batches : ',n(2)
         write (*,*) ' Batch size               : ',n(3)
         write (*,*) ' Quantile                 : ',p
         write (*,*) ' Confidence               : ',g
         write (*,*) ' Trickett-Welch iteration : ',int(xdx(i))
         write (*,*)
c
         do 60 j=1, nrho-1
c
            rho = (j-1) *dr
            r   = rho /(1. -rho)
c
c      -- Varaince components and mean corresponding to tolerance
c         limit factor.  The mean is taken to equal the percentile.
            s2w = 1 /(1 +r)
            s2b = r *s2w
            xmu = anorin (p) *sqrt (s2w +s2b)
c
c      -- Subroutine to evaluate the coverage probability
            call tlmcdf (xmu,s2b,s2w,n,p, 0., cov(i), file2)
c
c      -- Write out intrclass correlation and coverage probability.
            write (*,*) rho,  cov(i)
60       continue
40    continue
c
      stop
      end
      program pltcdf
c     ----------------------------------------------------------------
      parameter (maxpts = 500, maxcrv = 50)
c
c      Mark Vangel, Oct. 1988
c
c         Program to calculate and plot the distribution of a tolerance
c     limit for a random effects model.
```

```
c
      logical       known
      character*20 flenme
      character*1  ans
      dimension  cdf (maxpts), quant (maxpts), ipoint (maxcrv),
     $           crossx(2), crossy(2), n(3)
      common /kw/ known, rho, xknown

c
      ipoint (1) = 0
      write (*,*) ' number of points per plot ?'
      read  (*,*)   nq
      nplot      = 0
c
c      -- Get parameters which are constant over plots
c         ('nfix' = number of fixed effects in nested model)
      write (*,*) ' nfix, i, j ?'
      read  (*,*)   n
      write (*,*) ' p ?'
      read  (*,*)   p
      write (*,*) ' confidence ?'
      read  (*,*)   g
      nfix = n (1)
      k    = n (2)
      l    = n (3)
c
c      -- Loop over all k-factor files.
      write (*,*) ' 0=brief output, 1=complete output ?'
      read  (*,*)   ibrief
1     continue
c
c      -- Specify a filename for tolerance limit factor
c         (output from program TRICK)
         write (*,*)     ' Filename (''k'' if rho is known) ?'
         read  (*,'(a20)') flenme
         if (flenme .eq. ' ') go to 2
         write (*,*)     ' Rho ?'
         read  (*,*)       rho
         if (rho .eq. 1) rho = rho -1.e-6
c
c      -- Intraclass correlation known
         if (flenme .eq. 'k') then
            known  = .true.
            call init (p, g, n)
         else
            known  = .false.
         end if
c
c      -- Var. ratio, var. components
         r   = rho /(1 -rho)
         s2w = 1. /(1 +r)
         s2b = r *s2w
         xmu = anorin (p) *sqrt (s2w +s2b)
c
c      -- Pointer to next plot
         nplot = nplot +1
         ipoint (nplot+1) = ipoint (nplot) +nq
c
c      -- Calculate cdf at equally spaced points within specified range
         write (*,*) ' Range of values for cdf ?'
         read  (*,*)   qmin, qmax
         dq = (qmax-qmin)/(nq-1.)
c
c      -- Header for plot
```

```
      write (*,*) ' Distribution of ',100*(1-p), ' percentile '
      write (*,*) '     from ', qmin, ' to ', qmax
      write (*,*) ' Mean = ',xmu, ' Var. components = ',s2b, s2w
      write (*,*) ' Tolerance limit factor file = ',flenme
      write (*,*) ' Number of groups, group size = ', k, l
      write (*,*) ' Confidence level = ', g
      write (*,*)
c
c    -- Calculate points on cdf
      do 20 i=1, nq
         idx          = (nplot -1)*nq +i
         quant (idx)  = (i-1) *dq +qmin
         call tlmcdf (xmu,s2b,s2w,n,p,quant(idx),cdf(idx),flenme)
         if (ibrief .eq. 1) then
            write (*,*) quant (idx), cdf (idx)
         end if
20    continue
      go to 1
c
c    -- Now plot the results
2     continue
      write (*,*) 'Plots ?'
      read  (*,'(a1)') ans
      if (ans .eq. 'y') then
         write (*,*) ' Min and max for abscissa ?'
         read  (*,*)  qmin, qmax
         write (*,*) ' Min and max for ordinate [0,0 for default] ?'
         read  (*,*)  omin, omax
c
c    -- Initialize PLOT-10 graphics
         call initt (960)
         call binitt
c
c    -- Set coordinate ranges
         call comset (ibasex(11), qmin)
         call comset (ibasex(12), qmax)
         if (omax .ne. 0.) then
            call comset (ibasey(11), omin)
            call comset (ibasey(12), omax)
         end if
c
c    -- Produce the plots
         call npts  (nplot *nq)
         call check (quant, cdf)
         call npts  (nq)
         call dsplay (quant, cdf)
         do 30 i=1, nplot-1
           call cplot (quant (i*nq+1), cdf (i*nq+1))
30       continue
c
c    -- Plot crosshairs at quantile and nominal coverage probability
         crossx (1) = xmu -anorin (p) *sqrt (s2w +s2b)
         crossx (2) = crossx (1)
         crossy (1) = 0.
         crossy (2) = 1.
         call npts (2)
         call cplot (crossx, crossy)
         crossx (1) = qmin
         crossx (2) = qmax
         crossy (1) = g
         crossy (2) = g
         call cplot (crossx, crossy)
c
c    -- Hardcopy option
```

```
         call scursr (ans, i1, i2)
         if (ans .eq. 'p') then
            call hdcopy
         end if
         call finitt (-1)
         go to 2
      end if
c
      stop
      end
      program trick
c     ----------------------------------------------------------------
c
c      Mark Vangel, June 1989
c
c         Program to calculate one sided tolerance limits by the
c      modified Trickett-Welch method.
c
      logical restrt
      dimension n(3)
      character*20 flenme, rsfile
      common /crs/   restrt, rsfile
c
c      -- Restart capability allows restarting from a previously
c         computed k function.
      write (*,*) 'Restart (1 or 0) ?'
      read  (*,*) ires
      if (ires .eq. 1) then
         restrt = .true.
         write (*,*) 'Restart file ?'
         read  (*,'(a20)') rsfile
      else
         restrt = .false.
      end if
c
c      -- Problem parameters
      write (*,*) 'Number of steps for ratio of mean squares ?'
      read  (*,*)  nstp
      write (*,*) 'Filename for k-function files ?'
      read  (*,'(a20)') flenme
      write (*,*) 'Number of iterations ?'
      read  (*,*)  niter
      write (*,*) '(1-quantile) for tolerance limit ?'
      read  (*,*)  p
      write (*,*) 'Confidence coefficient for lower limit ?'
      read  (*,*)  g
      write (*,*) 'Number of fixed effects (for nested model) ?'
      read  (*,*)  n (1)
      write (*,*) 'Number of random batches ?'
      read  (*,*)  n (2)
      write (*,*) 'Batch size ?'
      read  (*,*)  n (3)
      call inteqn (p, g, n, nstp, flenme, niter, 1.)
      stop
      end
```

```
      subroutine evcdf (cum, idf1a, idf2a, c1a, c2a, xncpa, etaa, serr)
c     ----------------------------------------------------------------
c
c      Mark Vangel, Oct. 1988
c
c      Routine called by 'TLMCDF'.
c
      double precision aerr, error, xl, xh, result, fncz
      external fncz
c
c      -- Parameters for FNCZ
      common /b1/ idf1, idf2, c1, c2, xncp, con, eta
      data hf /.5/
c
c      -- Double precision error
      aerr  = serr
c
c      -- Integration rule
      irule = 2
c
c      -- Put stuff in common
      idf1  = idf1a
      idf2  = idf2a
      eta   = etaa
      c1    = c1a
      c2    = c2a
      xncp  = xncpa
      con   = alngam(hf*(idf1+idf2)) -alngam(hf*idf1) -alngam(hf*idf2)
c
c      -- Limits of integration. Avoid zero and one.
      xl    = 1.d-10
      xh    = 1.d0 - 1.d-10
c
c      --  Do the integration.
      call dqdag  (fncz, xl, xh, aerr, 0.d0, irule, result, error)
      cum   = result
c
      return
      end
      real function fnck (x)
c     ----------------------------------------------------------------
      real*8 tnc
      common /kcom/ xncp, g, idf, df
c
c      Called by root finder in 'INVNCT'.
c
c     --  Noncentral t with non-integer degrees of freedom
c         (Satterthwaite d.f. need not be an integer)
      fnck = tnc (dble(x), dble(df), dble(xncp), ifault) -g
      return
      end
      real function fncn (x)
c     ----------------------------------------------------------------
c
c      Called by root finder in 'INVSPN'.
c
      common /sp2/ y
      call kspln (x, y1)
      fncn = y -y1
      return
      end
      real function fncr (x)
c     ----------------------------------------------------------------
c
```

```
c     Mark Vangel, June 1986
c
c        Routine called by root finder in subroutine 'KR'.
c
      common /kr1/ c1, c2, idf1, idf2, xkp, xkg, p, g, xncp
      fncr = g -gends2 (x, idf1, idf2, c1, c2, xncp, 1.e-7)
      return
      end
      real function fncs (x)
c     -------------------------------------------------------------
c
c     Called by maximization routine in 'FSUP'.
c
      double precision xint
      common /ca/ nfix, k, l, eta, xkp, xkg, idrv, con
c
      idrv = 1
      fncs = -xint (dble (x))
      return
      end
      double precision function fncy (f)
c     -------------------------------------------------------------
c
c      Mark Vangel, June 1986
c
c         Called by numerical integration subroutine in 'GENDS2'.
c
      implicit real (a-h, o-z)
      double precision f
      common /b1/ idf1, idf2, tval, c1, c2, xncp, con
      data hf /0.5/
      fncy  = dble ((hf*idf2-1) *log (f)   +(hf*idf1-1) *log (1 -f))
      arg   = tval *sqrt (c1*(1. -f) +c2*f)
      if (arg .gt. 1.e10) then
         tprob = 1.
      else if (arg .lt. -1.e10) then
         tprob = 0.
      else
         tprob = tndf (arg, idf1+idf2, xncp)
      end if
      fncy  = dble(exp(fncy +con) *tprob)
      return
      end
      double precision function fncz (f)
c     -------------------------------------------------------------
c
c     Called by 'DQDAG' in 'EVCDF'.
c
      implicit real (a-h, o-z)
      double precision f
      common /b1/ idf1, idf2, c1, c2, xncp, con, eta
      data hf, one, zero /.5, 1., 0./
c
      fncz = (hf*idf2-one) *dlog (one -f) +(hf*idf1-one) *dlog (f)
      x    = eta *idf2 *f /((idf1 *(one -f)))
c
c      -- Spline interpolation of tolerance limit factor
      call kspln (x, xk)
      arg  = xk *sqrt (c1 *f +c2 *(one -f))
      if (arg .gt. 1.e10) then
         tprob = one
      else if (arg .lt. -1.e10) then
         tprob = zero
      else
```

```
         tprob = tndf (arg, idf1+idf2, xncp)
      end if
      fncz  = dble (exp (fncz +con) *tprob)
c
      return
      end
      real function gends2 (tvala,  idf1a, idf2a, c1a,
     $                                     c2a,    xncpa, serr)
c     ----------------------------------------------------------------
c
c     Mark Vangel, June, 1986
c
c        Evaluate generalized non-central t using integral
c     representation.
c
c      tvala          -- Argument of gen nct
c      idf1a, idf2a   -- Degrees of freedom for chisquares
c      c1a, c2a       -- Corresponding coefficients
c      xncpa          -- Noncentrality parameter
c      serr           -- Absolute error for num. integration
c
      implicit real (a-h, o-z)
      double precision aerr, error, xl, xh, result, fncy, rerr
      external fncy
C
C      -- Constants for fncy
      common /b1/ idf1, idf2, tval, c1, c2, xncp, con
c
c      -- Constants for common block
      idf1 = idf1a
      idf2 = idf2a
      tval = tvala
      c1   = c1a
      c2   = c2a
      xncp = xncpa
c
c      -- Constants for numerical integration
      aerr = serr
      rerr = 0.d0
      hf   = 0.5
      xl = 1.d-10
      xh = 1.d0  - xl
      con  = alngam(hf*(idf1+idf2)) -alngam(hf*idf1) -alngam(hf*idf2)
      call dqdags (fncy, xl, xh, aerr, rerr, result, error)
      gends2 = sngl (result)
      return
      end
      subroutine inispl (lfn)
c     ----------------------------------------------------------------
c
c      Mark Vangel, Oct. 1988
c
c         Routine to initialize a spline fit to data read from
c      a file.  The unit number of the file is 'lfn'.  The tolerance
c      limit files are output from program 'TRICK'.
c
      common /cb/ xs (500), xks (500), break (500), c (4, 500), nx
c
      read (lfn, *) nx
      do 10 i=1, nx
         read (lfn, *) xs (i), xks (i)
10    continue
c
      call csint (nx, xs, xks, break, c)
```

```
c
      return
      end
      subroutine init (p, g, n)
c     ----------------------------------------------------------------
c
c      Mark Vangel, Oct. 1988
c
c         Initialize a few constants that will be needed later.
c
      logical known, meowen
      dimension n (3)
      common /ca/ nfix, k, l, eta, xkp, xkg, idrv, con
      common /kw/ known, rho, xknown
      common /cc/ xkzero, xkinf
      common /cd/ diag
      data one /1./, half /0.5/, rhoeps /1.e-4/
c
      nfix = n (1)
      k    = n (2)
      l    = n (3)
c
c        -- Normal quantiles.
      xkp  = anorin (p)
      xkg  = anorin (g)
c
c        -- Log beta function.
      df1  = nfix *(k -one)
      df2  = nfix *(k *(l -one))
      con = alngam (half *(df1 +df2)) -
     $      alngam (half *df1) -alngam (half *df2)
c
c     -- k values for zero and infinity
      call kr (n, p, g, 1., xkzero)
      call kfact (p, g, k-1, xkinf)
      if (diag .eq. one) write (*,*) 'init : xkzero, xkinf ',
     $                  xkzero, xkinf
c
c      -- If rho is known, calculate constant k value; using limit for
c         rho = one
      if (known  .and. abs (rho -one) .gt. rhoeps) then
         t  = l *rho /(one -rho) +one
         call kr (n, p, g, t, xknown)
      else if (known) then
         xknown = xkinf
      end if
c
      return
      end
      subroutine inteqn (pa, ga, n, nstpa, flenme, niter, diaga)
c     ----------------------------------------------------------------
c
c      Mark Vangel, Oct. 1988
c
c         Top-level subroutine to compute tolerance limits by a
c      modified Trickett-Welch procedure.
c
c      p       --   Probability associated with quantile
c      g       --   Confidence associated with tolerance limit
c      n (1)   --   Number of fixed factor levels  (nfix)
c      n (2)   --   Number of batches              (k)
c      n (3)   --   Batch size                     (l)
c      nstp    --   Number of steps for nuisance parameter
c      flenme --   Filename for output of tolerance limit factor
```

```
c                     estimates
c       niter  --    Number of iterations
c       diag   --    =1 for debug lines to print
c
      parameter  (maxpts = 500)
      logical        known, restrt
      character*20 flenme, rsfile
      character*30 file2
      character*3  citer
c
      dimension xk      (maxpts),    xk1 (maxpts), cvrate (maxpts),
     $           xkstep (maxpts),
     $           x       (maxpts),    n(3)
c
      common /crs/ restrt, rsfile
      common /kw/ known, rho, xknown
      common /ca/ nfix, k, l, eta, xkp, xkg, idrv, con
      common /cc/ xkzero, xkinf
      common /cd/ diag
c
      data zero /0./,  one /1./, two  /2./ ,  lfn /10/, rh /20./
c
c      -- Initialization
      known = .false.
      nfix  = n (1)
      k     = n (2)
      l     = n (3)
      p     = pa
      g     = ga
      nstp  = nstpa
      diag  = diaga
c
c      -- Degrees of freedom
      df1   = nfix *(k -one)
      df2   = nfix *(k *(l -one))
c
c      -- Constants for repeated use
      call init (p, g, n)
c
c      -- Initial step size for x
      dx                 = (l*rh +one) /(nstp -one)
      iter               = 0
c
c      -- First guess at value at zero. Note that Welch result
c         blows up at zero, hence it can't be used here.
      x      (1)       = zero
      xk     (1)       = xkzero
      cvrate (1)       = g
c
c      -- Values at infinity : exact
      x      (nstp +1) = one
      xk     (nstp +1) = xkinf
      cvrate (nstp +1) = g
c
c      -- Option to continue a previous calculation
      if (restrt) then
         open (unit=10, file=rsfile)
         do 20 i=1, nstp +1
           read (10, *) x (i), xk (i)
20       continue
      else
c
c      -- First pass at Welch series : equally spaced abcissas
         do 21 i=2, nstp
```

```
          x  (i)      = (i -one) *dx
          xk (i)      = welch (x (i), xkg, xkp, n)
21      continue
c
c    -- Write out first pass to a scratch file for 'mesh'
        open (unit=lfn, status='scratch' )
        write (lfn, *) nstp
        do 22 i=1, nstp +1
            write (lfn, *) x (i), xk (i)
22      continue
c
c    -- Find mesh which gives equal spacing in y
        rewind      (lfn)
        call inispl (lfn)
        call mesh (x)
        close       (lfn)
c
c    -- Second pass at Welch series : equally spaced ordinates
        do 23 i=2, nstp
          xk (i)      = welch (x (i), xkg, xkp, n)
23      continue
c
      end if
c
c    -- Loop over iterations
      do 30 i=1, niter
c
c    -- Filename for output.  Iteration number appended to name.
        write (unit=citer, fmt='(a1,i2)') '-',i
        if (citer (2:2) .eq. ' ') citer (2:2) = '0'
        lstnbk = 20
31      continue
          if (flenme (lstnbk:lstnbk) .eq. ' ') then
            lstnbk = lstnbk -1
            go to 31
          end if
        file2 = flenme (1:lstnbk) //citer
c
c    -- Write out latest results to file
        open (unit=lfn, file=file2, status='new')
        write (lfn, *) nstp
        do 40 j=1, nstp +1
            write (lfn, *) x (j), xk (j)
40      continue
c
c    -- Initialize the spline with latest results
        rewind (lfn)
        call inispl (lfn)
c
c    -- Use Trickett-Welch to get improved approximation to xk
        call nextk (n, nstp, x, xk, xk1, xkstep, cvrate, p, g)
c
c    -- Rewrite the current approximation  with coverage rates
        rewind (lfn)
        write  (lfn, *) nstp
        do 50 j=1, nstp +1
            write (lfn, *) x (j), xk (j), cvrate (j)
50      continue
c
c    -- Update current approximation
        do 60 j=1, nstp
          xk (j) = xk1 (j)
60      continue
c
```

```
c        -- Now do it all over again ...
           iter = iter +1
30     continue
c
       return
       end
       subroutine invnct (ga, dfa, xncpa, xl, xh, t)
c      ----------------------------------------------------------------
c
c       Mark Vangel, Dec. 1988
c
c          Subroutine to invert the noncentral t distribution. the
c       limits 'xl' and 'xh' contain the root and are input parameters.
c
       common /kcom/ xncp, g, idf, df
       external fnck
       data aerr /1.e-5/, rerr /1.e-5/
c
       g     = ga
       xncp  = xncpa
       df    = dfa
       idf   = df
       a     = xl
       t     = xh
       maxfn = 250
       call zbren (fnck, aerr, rerr, a, t, maxfn)
       return
       end
       subroutine invspn (xla, xha, ya, x)
c      ----------------------------------------------------------------
c
c      Mark Vangel, Dec. 1988
c
c          Subroutine 'INVSPN' performs inverse spline interpolation.
c      This routine is called by 'MESH"
c
       common /sp2/ y
       external fncn
       data zero /0.0/, eps /1.e-5/
c
       xl = xla
       xh = xha
       y  = ya
c
       errrel = eps
       errabs = zero
       maxfn  = 100
       call zbren (fncn, errabs, errrel, xl, xh, maxfn)
       x      = xh
       return
       end
       subroutine kfact (p, g, idf, xk)
c      ----------------------------------------------------------------
c
c       Mark Vangel, June 1986
c
c          Subroutine to compute tolerance limit factor for a
c       simple normal sample.
c
       data one /1./, uplim /25./
c
       xncp  = anorin (p) *sqrt (idf +one)
       call invnct (g, float(idf), xncp, one, uplim, t)
       xk    = t /sqrt (idf +one)
```

```
c
      return
      end
      subroutine kmo (i, j, p, g, xmsr, fp, xk)
c     ---------------------------------------------------------------
c
c      Mark Vangel, Dec. 1988
c
c      Calculate the Mee-Owen tolerance limit factor.
c
      data zero /0/, half /.5/, one /1.0/, xh /25/
c
c      -- upper confidence bound on variance ratio
      df1      = i -one
      df2      = i *(j-one)
      fconf    = fin (fp, df2, df1)
      r        = (xmsr*fconf -one) /real (j)
c
c      -- noncentrality parameter
      xb       = (r +one) / (j*r +one)
      xncp     = sqrt (i*j *xb) *anorin(p)
c
c      -- Satterthwaite degrees of freedom
      sdf    = (r +one)**2 /
     &         ((r+one/j)**2/(i-1) +(one-one/j)/(i*j))
c
c      -- noncentral t quantile
      call invnct (g, sdf, xncp, one, xh, xk)
c
c     -- tolerance limit factor
      xk = xk /sqrt (i*j *xb)
c
      return
      end
      subroutine kr (n, pa, ga, teta, xk)
c     ---------------------------------------------------------------
c
c      Mark Vangel, June 1986
c
c         Routine to determine tolerance limit factors for known
c     variance ratio r.
c
c         n       --   number fixed effects, batches, batch size
c         pa      --   quantile
c         ga      --   confidence for lower tolerance limit
c         teta    --   eta=j*r+1 (known) nuisance parameter
c         xk      --   returned tolerance limit factor
c
      dimension n (3)
      external fncr
      common /kr1/ c1, c2, idf1, idf2, xkp, xkg, p, g, xncp
c
c      -- Constants for common block
      eta = teta
      p      = pa
      g      = ga
      xkg    = anorin (g)
      xkp    = anorin (p)
      if (eta .eq. 0.) eta = .05
c
c      -- Degrees of freedom
      nfix   = n (1)
      i      = n (2)
      j      = n (3)
```

```
      idf1  = nfix *(i-1)
      idf2  = nfix *(i*(j-1))
c
c      -- Coefficients for linear comb. of chi-squares
      c1    = i*j -1.
      c2    = c1 /eta
      c1    = c1 *i/(i-1.)
c
c      -- Noncentrality parameter
      xncp  = xkp *sqrt (i*(1. +(j-1.)/eta))
c
c      -- Root finder
      aerr  = 1.e-5
      rerr  = 1.e-5
      a     = 1.
      b     = 10. *xncp
      maxfn = 100
      call zbren  (fncr, aerr, rerr, a, b, maxfn)
      xk    = b
      return
      end
       subroutine kspln (eta, xk)
c     ------------------------------------------------------------------
c
c      Mark Vangel, Oct. 1988
c
c          Spline interpolation of tolerance limit factor.
c
      logical known
      dimension m(3)
      common /ca/ nfix, k, l, deta, xkp, xkg, idrv, con
      common /cb/ xs (500), xks (500), break (500), c (4, 500), nx
      common /kw/ known, rho, xknown
      data iord /10/, one /1.0/
c
c      -- Use constant value if rho is known
      if (known .and. rho .ge. 0) then
           xk = xknown
c
c      -- Truncate function at upper limit calculated
      else if (eta .gt. xs(nx)) then
            xk = xks (nx)
      else
c
c      -- Spline interpolation
         xk = csval (eta, nx-1, break, c)
      end if
      return
      end
      subroutine mesh (xnew)
c     ------------------------------------------------------------------
      common /cb/ xs (500), xks (500), break (500), c (4, 500), nx
c
c     Mark Vangel, Dec. 1988
c
c         The initial mesh of abscissa values is taken to be equally
c     spaced.  A better approximation can be obtained if more points
c     are taken where the function being estimated changes most
c     rapidly, however.  Subroutine 'MESH' takes as input the
c     initial equally spaced mesh and the Welch approximation at
c     these mesh points.  The ordinate is equally divided into
c     intervals and the Welch approximation provides (via inverse
c     interpolation in 'INVSPN') the corresponding new mesh of
c     abscissa values.  This new mesh is used for all succeeding
```

```
c     iteration.
c
      dimension xnew (1)
      data one /1./
c
      dlty = (xks(nx) -xks(1)) /(nx-one)
      do 10 i=2, nx-1
         y = (i-1) *dlty +xks(1)
         call invspn (xs(1), xs(nx), y, xnew(i))
10    continue
      xnew(1)  = xs(1)
      xnew(nx) = xs(nx)
      return
      end
      subroutine nctd1n (idf, tval, xncp, densty)
c     -----------------------------------------------------------------
c
      p1 = tndf (sqrt((idf+2.)/idf) *tval, idf+2, xncp)
      p2 = tndf (tval,                     idf,   xncp)
c
      densty = (idf/tval) *(p1 -p2)
      return
      end
      subroutine nctd2n (idf, tval, xncp, drv)
c     -----------------------------------------------------------------
c
      c = sqrt ((idf+2.)/idf)
      call nctd1n (idf+2, c*tval, xncp, p1)
      call nctd1n (idf  . tval,   xncp, p2)
      drv = (idf/tval) *(p1 -p2)
      return
      end
      subroutine nctd3n (idf, tval, xncp, drv)
c     -----------------------------------------------------------------
c
      c = sqrt ((idf+2.)/idf)
      call nctd2n (idf+2, c*tval, xncp, p1)
      call nctd2n (idf,   tval,   xncp, p2)
      drv = (idf/tval) *(p1 -p2)
      return
      end
      subroutine nctd4n (idf, tval, xncp, drv)
c     -----------------------------------------------------------------
c
      c = sqrt ((idf+2.)/idf)
      call nctd3n (idf+2, c*tval, xncp, p1)
      call nctd3n (idf,   tval,   xncp, p2)
      drv = (idf/tval) *(p1 -p2)
      return
      end
      subroutine nctd5n (idf, tval, xncp, drv)
c     -----------------------------------------------------------------
c
      c = sqrt ((idf+2.)/idf)
      call nctd4n (idf+2, c*tval, xncp, p1)
      call nctd4n (idf,   tval,   xncp, p2)
      drv = (idf/tval) *(p1 -p2)
      return
      end
      subroutine nctdrv (k, idf, tval, xncp, drv)
c     -----------------------------------------------------------------
c
c      Mark Vangel, May 1986
c
```

```
c          Evaluate either the noncentral t cumulative (k=0) or
c       the kth derivative of the cumulative with respect to the
c       argument (k=1,2,3,4,5).  derivatives are calculated exactly
c       in terms of the cumulative by means of a recursion formula.
c
      if (k .eq. 0) then
         drv = tndf (tval, idf, xncp)
      else if (k .eq. 1) then
         call nctd1n (idf, tval, xncp, drv)
      else if (k .eq. 2) then
         call nctd2n (idf, tval, xncp, drv)
      else if (k .eq. 3) then
         call nctd3n (idf, tval, xncp, drv)
      else if (k .eq. 4) then
         call nctd4n (idf, tval, xncp, drv)
      else if (k .eq. 5) then
         call nctd5n (idf, tval, xncp, drv)
      end if
      return
      end
      subroutine nextk
     $       (n, nstp, x, xk0, xk1, xkstep, cvrate, pa, ga)
c     -----------------------------------------------------------------
c
c      Mark Vangel, Oct. 1988
c
c         Given an input tolerance limit factor xk0 and the parameters
c      of the problem, this subroutine calculates the next approximation
c      xk1 by a modified Trickett-Welch procedure.
c
c        n (1)    --   Number of fixed factor levels  (nfix)
c        n (2)    --   Number of batches              (k)
c        n (3)    --   Batch size                     (l)
c         nstp    --   Number of intervals for k-function
c         x       --   Values of nuisance parameter
c         xk0     --   Input k-function
c         xk1     --   Output k-function
c         cvrate --    Coverage probability of xk1
c         p       --   Probability level of quantile
c         g       --   Confidence level of tolerance limit
c
      dimension x(1), xk0 (1), xk1 (1), xkstep (1), cvrate (1), n(3)
      double precision tl, th, daerr, drerr, xi0, xi1, errest
      common /ca/ nfix, k, l, eta, xkp, xkg, idrv, con
      common /cd/ diag
      common /sblk/ oldsup
      external xint
      data tl /1.d-5/, th /1.d0/, aerr/0.0/, rerr/1.e-4/, one /1./,
     $     zero /0./, daerr /1.d-5/, drerr /0.d0/
c
c     -- Initialize some constants
      th    = one  -tl
      nfix  = n (1)
      k     = n (2)
      l     = n (3)
      p     = pa
      g     = ga
      df1   = nfix *(k -1)
      df2   = nfix *(k*(l -1))
      irule = 2
c
c      -- Find peak of integrand and determine transformation
      x0 = sup (float(20), ier)
      if (ier .ne. 0) x0 = oldsup
```

```
      if (x0 .eq. zero) x0 = df1 /(df1 +df2)
      oldsup = x0
      alpha =  (df1 /df2) *(one -x0) /x0
      write (*,*) ' alpha = ', alpha
c
      do 10 i=1, nstp
         eta   =  alpha*x(i) +one
c
c      -- First integral
         idrv  = 0
         call dqdag (xint, tl, th, daerr, drerr, irule, xi0, errest)
c
c      -- Second integral (derivative)
         idrv  = 1
         call dqdag (xint, tl, th, daerr, drerr, irule, xi1, errest)
         cvrate (i) = xi0
         xkstep (i) = (g -cvrate (i)) /xi1
         xk1    (i) = xk0 (i) +xkstep (i)
         if (diag .eq. one) write (*,*) 'nextk : k, cvrate, step ',
     $      i, x (i), xk0 (i), cvrate (i), xkstep (i)
10    continue
c
      return
      end
      real function sup (r, ier)
c     ---------------------------------------------------------------
c
c      Mark Vangel, Dec 1988
c
c         Find the maximum of Trickett-Welch integrand for
c      variance ratio equal to r.  The spline for XK must be
c      initialized before this routine may be used.  Also, the
c      stuff in /ca/ must be available.
c
      common /ca/ nfix, k, l, eta, xkp, xkg, idrv, con
      external fncs
      data one /1./, eps /1.e-5/, xacc /.001/
      eta = l*r +one
c
c      -- find minimum by Brent's method
      xguess = one /2
      bound  = xguess -eps
      xstep  = one /4
      maxfn  = 100
      call uvmif (fncs, xguess, step, bound, xacc, maxfn, peak)
      sup = peak
c
      return
      end
      subroutine tlmcdf (xmu, s2b, s2w, n, p, t, cum, flenme)
c     ---------------------------------------------------------------
c
c      Mark Vangel, Oct. 1988
c
c         Cumulative distribution function of the lower confidence
c      bound on s quantile from a random effects model.  This routine
c      can be used with a nested model.  The results in the paper
c      correspond to nfix=1.
c
c      xmu    --  population mean
c      s2b    --  population variance between groups
c      s2w    --  population variance within groups
c      nfix   --  number of groups
c      i      --  number of batches
```

```
c      j      --  batch size
c      p      --  probability associated with quantile
c      t      --  value at which cdf is to be evaluated
c      cum    --  probability tol. limit. is less than pth quantile
c      flenme --  name of ASCII file containing tolerance limit factor
c                       row 1    :  number of steps
c                       row 2..n :  msr /(msr +1), k-factor
c                                        row 2 : msr = 0
c                                        row n : msr = infinity
c                       (output from porgram TRICK)
c
c
      logical known
      character*20 flenme
      dimension n (3)
c
c      -- Spline for tolerance limit factor
      common /cb/ xs (500), xks (500), break (500), c (4, 500), nx
c
c      -- Flag set if rho taken to be known
      common /kw/ known, dummy, xknown
      data lfn /10/
c
c      -- Initialize spline for tolerance limit factor
      if (.not. known) then
         open (unit=lfn, file=flenme, iostat=lstat)
         rewind      (lfn)
         call inispl (lfn)
      end if
      nfix  = n (1)
      i     = n (2)
      j     = n (3)
c
c      -- Set up parameters
      aerr  = 1.e-5
      xkp   = anorin (p)
      eta   = j *s2b /s2w +1.
      idf1  = nfix *(i-1)
      idf2  = nfix *(i *(j -1))
      c1    = (idf1 +idf2) /nfix
      c2    = c1 /eta
      c1    = c1 *i /(i -1.)
      xncp  = (xmu -t) /sqrt ((j*s2b +s2w) /(i*j))
c
c     -- Evaluate cdf of lower tolerance limit
      call evcdf (cum, idf1, idf2, c1, c2, xncp, eta, aerr)
c
      return
      end
      real function welch (ymsr, xkg, xkp, m)
c     ---------------------------------------------------------------
c
c      Mark Vangel, July 1986
c
c         First order Welch-type expansion for the tolerance
c     limit factor.
c
      real i, l
      dimension m (3)
c
      i = m(1) *(m(2)-1) +1
      l = real (m(1) *m(2) *(m(3)-1)) /
     $    real (m(1) *(m(2)-1) +1) +1
c
```

```
      xmsr  =  ymsr
      n   = i*l
      t1  = sqrt (1/(1+(l-1)/xmsr))
      t2  = sqrt (1/(xmsr*xmsr +(l-1)*xmsr))
      rti = sqrt (i)
      rtn = sqrt (float(n))
      xl1 = 1/ (l*l)
      xl2 = ((l-1)/l) **2
c
      xk  = xkp +t1/rtn *(xkg +1./(4*(i-1)) *(
     $           xkg *(xkg*xkg +1)          +xkp*xkp*xkg *n *t1*t1 *xl1
     $          +xkp *rtn *t1*t1*t1 *xl1     +xkp*xkg*xkg *rtn *t1 /l)
     $                        +1/(4*i*(l-1.d0)) *(
     $          +xkp*xkp* xkg *n *t2*t2 *xl2  +xkp *rtn *t2*t2*t2 *xl2))
c
      welch = xk
      return
      end
        double precision function xint (x)
c     ----------------------------------------------------------------
c
c      Mark Vangel, Oct. 1988
c
c         Function to calculate two integrands needed for the
c      Trickett-Welch procedure.  One integrand is a noncentral
c      t cumulative 'weighted' by a beta density; the other
c      integrand is the derivative of this first integrand with
c      respect to the k-function (which is part of the argument
c      of the noncentral t cumulative).
c
      double precision x
      common /ca/ nfix, k, l, eta, xkp, xkg, idrv, con
      data one /1./, half /0.5/, zero /0.0/, tiny /1.e-6/
c
c      -- Between, within, total degrees of freedom
      df1  = nfix *(k -1)
      df2  = nfix *(k *(l -1))
      df   = df1 +df2
c
c      -- Calculate mean square ratio.  Use asymptote when mean
c         square ratio is infinite
      x1 = x
      if (x1 .le. zero) x1 = tiny
      r = eta *df2 *x1 / (df1 *(one -x1))


c
c      -- Cubic spline interpolation of k-function
      call kspln (r, xk)
c
c      -- Noncentrality parameter and argument for noncentral t
      xncp = xkp *sqrt (k*(one +(l -one) /eta))
      arg  = sqrt (df /nfix *(k/(k -one) *x1 +(one -x1) /eta))
c
c     -- This subroutine can calculate higher derivatives than
c        the first if desired.
      arg1 = one
      fact = one
      do 10 i=1, idrv
         arg1 = arg1 *arg
         fact = fact *i
         arg1 = arg1 /fact
10    continue
c
c      -- Noncentral t cunulative or it's derivatives
```

```
          call nctdrv (idrv, int(df), xk *arg, xncp, prob)
c
c      -- Beta density
       if (x1 *(one -x1) .ne. zero) then
          beta  = (half *df1 -one) *log (x1)
     $           +(half *df2 -one) *log (one -x1)
       else if (x1 .eq. zero) then
          beta  = (half *df2 -one)  *log (one -x1)
       else
          beta  = (half *df1 -one) *log  (x1)
       end if
c
c      -- finally, return the integrand.
       xint = arg1 *prob *exp (beta +con)
c
       return
       end
```

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION Unclassified	1b. RESTRICTIVE MARKINGS
2a. SECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION/AVAILABILITY OF REPORT Unlimited
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE	
4. PERFORMING ORGANIZATION REPORT NUMBER(S) TR No. ONR-C-5	5. MONITORING ORGANIZATION REPORT NUMBER(S)
6a. NAME OF PERFORMING ORGANIZATION Department of Statistics Harvard University 6b. OFFICE SYMBOL (If applicable)	7a. NAME OF MONITORING ORGANIZATION
6c. ADDRESS (City, State, and ZIP Code) Dept. of Statistics Harvard University Cambridge, MA 02138	7b. ADDRESS (City, State, and ZIP Code)
8a. NAME OF FUNDING/SPONSORING ORGANIZATION ONR 8b. OFFICE SYMBOL (If applicable) Code 1111	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER N0001486-K-0246
8c. ADDRESS (City, State, and ZIP Code) Office of Naval Research Arlington, VA 22217-5000	10. SOURCE OF FUNDING NUMBERS PROGRAM ELEMENT NO. / PROJECT NO. / TASK NO. / WORK UNIT ACCESSION NO

11. TITLE (Include Security Classification)
The Trickett-Welch 'Solution' to the Behrens-Fisher Problem Applied to One-Sided Tolerance Limits for Random Effects Models

12. PERSONAL AUTHOR(S)
Mark G. Vangel

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (Year, Month, Day)	15. PAGE COUNT
Technical Report	FROM ____ TO ____	January 1990	56

16. SUPPLEMENTARY NOTATION

17. COSATI CODES		
FIELD	GROUP	SUB-GROUP

18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

See reverse Side

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT ☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT ☐ DTIC USERS	21. ABSTRACT SECURITY CLASSIFICATION
22a. NAME OF RESPONSIBLE INDIVIDUAL Herman Chernoff	22b. TELEPHONE (Include Area Code) 617-495-5462 22c. OFFICE SYMBOL

DD FORM 1473, 84 MAR

83 APR edition may be used until exhausted
All other editions are obsolete

SECURITY CLASSIFICATION OF THIS PAGE
UNCLASSIFIED